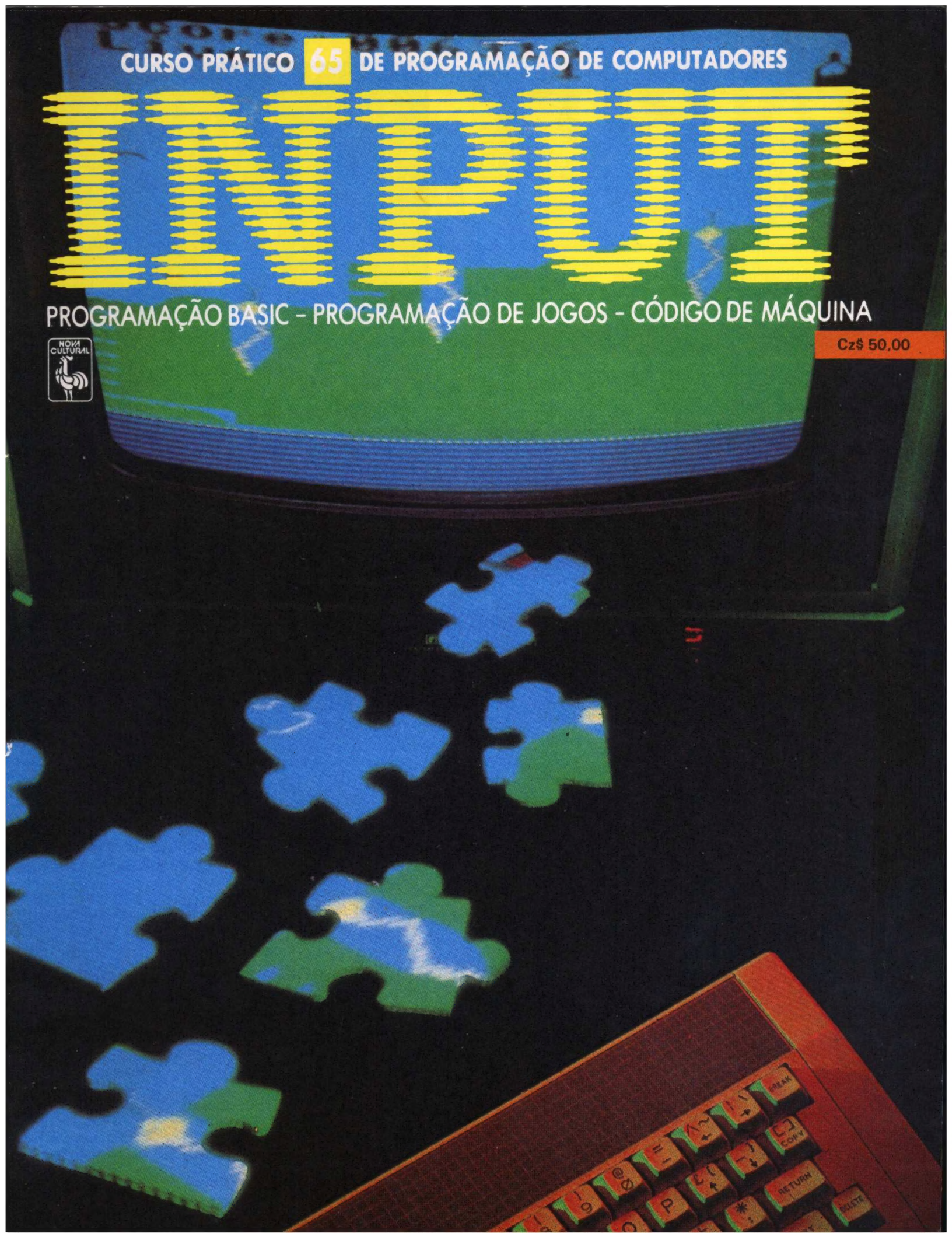
CURSO PRÁTICO 65 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

NOVA CULTURAL

Cz$ 50,00



INPUT

Vol. 5 N.º 65

NESTE NÚMERO

APLICAÇÕES

FERRAMENTAS PARA O SPECTRUM

Adição de comandos extras. Rotina de renumeração de linhas. Função de autonumeração de linhas. Eliminação em bloco. Outros comandos... 1281

PERIFÉRICOS

ROBÔS CONTROLADOS POR COMPUTADOR

Inteligência Artificial. Codificação binária. Tipos de robô. Linguagem de controle. Sentidos robóticos. A tartaruga de Papert 1284

LINGUAGEM

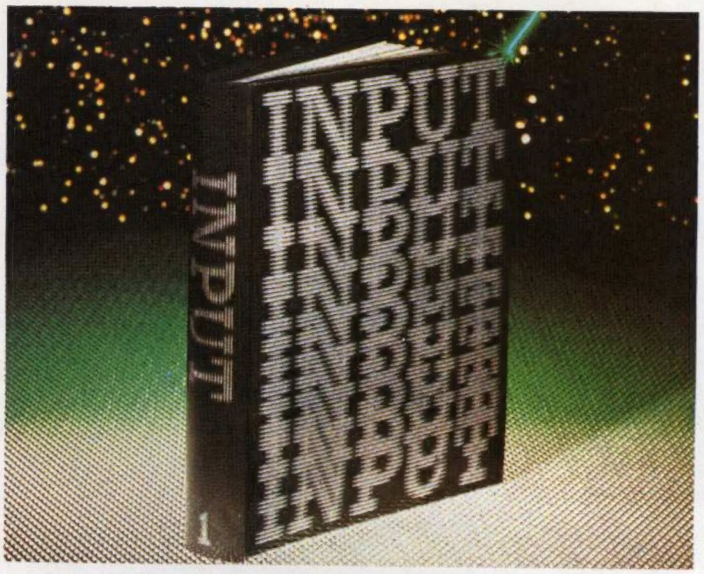
A TORRE DE BABEL

Desenvolvimento de novas linguagens. Raízes. Evolução. Linguagens imperativas e funcionais. As linguagens da quarta geração 1288

CÓDIGO DE MÁQUINA

AVALANCHE: LISTAGEM COMPLETA

Correção de erros. Teste do conjunto. O código hexadecimal. Alterações possíveis..... 1292



PLANO DA OBRA

"INPUT" é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

COMPLETE SUA COLEÇÃO

Exemplares atrasados, até seis meses após o encerramento da coleção, poderão ser comprados, a preços atualizados, da seguinte forma: 1. PESSOALMENTE — Por meio de seu jornaleiro ou dirigindo-se ao distribuidor local, cujo endereço poderá ser facilmente conseguido junto a qualquer jornaleiro de sua cidade. Em **São Paulo**, os endereços são: rua Brigadeiro Tobias, 773, Centro; avenida Industrial, 117, Santo André; e no **Rio de Janeiro**: avenida Mem de Sá, 191/193, Centro. 2. POR CARTA — Poderão ser solicitados exemplares atrasados também por carta, que deve ser enviada para DINAP — Distribuidora Nacional de Publicações — Números Atrasados — Estrada Velha de Osasco, 132, Jardim Teresa — CEP 06000 — Osasco — SP. Não envie pagamento antecipado. O atendimento será feito pelo reembolso postal e o pagamento, incluindo as despesas postais, deverá ser efetuado ao se retirar a encomenda na agência do Correio. 3. POR TELEX — Utilize o nº (011) 33 670 DNAP.

Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações, Lda. — Qta. Pau Varais, Azinhaga de Fetais — 2 685, Camarate — Lisboa; Apartado 57 — Telex 43 069 JARLIS P.

Atenção: Após seis meses do encerramento da coleção, os pedidos serão atendidos dependendo da disponibilidade do estoque.

Obs.: Quando pedir livros, mencione sempre título e/ou autor da obra, além do número da edição.

COLABORE CONOSCO

Encaminhe seus comentários, críticas, sugestões ou reclamações ao Serviço de Atendimento ao Leitor — Caixa Postal 9442, São Paulo — SP.

Editor
VICTOR CIVITA

REDAÇÃO
Diretor Editorial: Carmo Chagas

Editores Executivos: Antonio José Filho, Berta Sztark Amar

Editor Chefe: Paulo de Almeida
Editora de Texto: Ana Lúcia B. de Lucena
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Dagmar Bastos Sampaio, Grace Alonso Arruda, Monica Lenardon Corradi
Secretária de Redação/ Coordenadora: Stefania Crema
Secretários de Redação: Beatriz Hagström, José Benedito de Oliveira Damião, Maria de Lourdes Carvalho, Marisa Soares de Andrade, Mauro de Queiroz

COLABORADORES

Consultor Editorial Responsável: Dr. Renato M. E. Sabbatini (Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas, SP
Tradução, adaptação, programação e redação: Abílio Pedro Neto, Aluísio J. Dornellas de Barros, Marcelo R. Pires Therezo, Marcos Huascar Velasco, Raul Neder Porrelli, Ricardo J. P. de Aquino Pereira
Coordenação Geral: Rejane Felizatti Sabbatini

COMERCIAL
Diretor Comercial: Roberto Martins Silveira
Gerente Comercial: Flávio Maculan
Gerente de Circulação: Denise Maria Mozol

PRODUÇÃO
Gerente de Produção: João Stungis
Coordenador de Impressão: Atílio Roberto Bonon

Preparador de Texto/Coordenador: Eliel Silveira Cunha
Preparadores de Texto: Alzira Moreira Braz, Ana Maria Dilguerian, Levon Yacubian, Luciano Tasca, Maria Teresa Galluzzi, Maria Teresa Martins Lopes, Paulo Felipe Mendrone
Revisor/Coordenador: José Maria de Assis
Revisoras: Conceição Aparecida Gabriel, Isabel Leite de Camargo, Ligia Aparecida Ricetto, Maria de Fátima Cardoso, Nair Lucia de Britto
Paste-up: Anastase Potaris, Balduino F. Leite, Edson Donato

© Marshall Cavendish Limited 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986.

Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, n.º 2000 - 3.º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta na AM Produções Gráficas Ltda. e impressa na Divisão Gráfica da Editora Abril S.A.

FERRAMENTAS PARA O SPECTRUM

- COMANDOS EXTRAS
- RENUMERAÇÃO DE LINHAS
- AUTONUMERAÇÃO DE LINHAS
- ELIMINAÇÃO EM BLOCO
- OUTROS COMANDOS

Este conjunto de ferramentas de programação BASIC tornará seu trabalho mais fácil. Com ele, você terá acesso a diversos comandos inteiramente novos no Spectrum.

Embora todos os micros examinados em *INPUT* utilizem a mesma linguagem BASIC, você já deve ter percebido que há diversas variantes, ou "dialetos" da mesma. De fato, é muito raro encontrar um programa, mesmo curto, que possa ser executado sem modificações em qualquer linha de computadores. Com freqüência, trata-se apenas de uma variação na sintaxe dos comandos ou na maneira como eles são usados; nesse caso, a adaptação do programa não envolve maiores complicações. Algumas vezes, porém, constata-se que muitos dos comandos destinados a um computador simplesmente não existem para outros. Ainda que eles não sejam essenciais para a programação de muitas tarefas, não há dúvida de que, se fossem disponíveis, o trabalho do programador se tornaria bem mais fácil. Incluem-se nessa categoria os comandos para renumeração das linhas de um programa (**RENUM**), para eliminação de blocos de linhas (**DEL**) etc. Presentes em computadores de linhas mais recentes, como o MSX, esses comandos fazem muita falta nos micros da linha Sinclair.

O programa apresentado neste artigo adiciona vários comandos desse tipo ao interpretador dos microcomputadores da linha Sinclair Spectrum, facilitando o desenvolvimento de programas em BASIC. Os micros das linhas TRS-80, TRS-Color e MSX não precisam de um programa como este, pois já dispõem da maioria dos comandos necessários em seu interpretador BASIC.

O programa foi desenvolvido em código de máquina, de modo a coexistir, na memória, com o programa em BASIC e o interpretador.

São adicionados ao interpretador oito novos comandos e funções:

- renumeração de linhas;
- eliminação de um bloco de linhas;
- número de bytes livres na memória;
- comprimento de um programa;
- numeração automática de linhas;
- catalogador de fitas;
- conversão de hexa para decimal e vice-versa.

Todas essas funções são chamadas por intermédio do comando **RANDOMIZE USR**, seguido de um número, conforme mostramos mais adiante.

Uma vez que um novo comando tenha sido ativado, ele pedirá diversos parâmetros — tais como os números das linhas a serem apagadas — através de mensagens exibidas na tela.

O programa utiliza diversas rotinas em linguagem de máquina fornecidas em artigo posterior (*Referência Cruzada*). Assim, será necessário juntar os dois programas. As instruções para a execução dessa tarefa estão contidas dentro do programa aqui apresentado. Bastará, portanto, digitar o programa, acioná-lo com um comando **RUN** e, então, seguir a orientação dada pelas mensagens exibidas na tela. O programa informará, também, se você cometeu algum erro na digitação das linhas **DATA**, que contêm o programa em código de máquina, pois efetua automaticamente uma soma de verificação. Corrija todos os erros, antes de armazenar o programa em fita.

Os códigos conjuntos do referenciador e do extensor de comandos serão armazenados em fita sob a denominação **"TOOLKIT" CODE**.

Para carregar o programa, digite:

```
CLEAR 63488
LOAD "TOOLKIT" CODE
```

A rotina de renumeração de linhas é ativada pelo comando:

```
RANDOMIZE USR 63489
```

O programa pede que se informe o incremento de linhas a ser utilizado (um número entre 1 e 255). Para achar o comprimento de um programa BASIC residente na memória, use:

```
RANDOMIZE USR 63889
```

Para achar o número de bytes disponíveis na memória, digite:

```
RANDOMIZE USR 63860
```

A numeração automática de linhas (comando **AUTO**, em outros computadores) é ativada por:

```
RANDOMIZE USR 64154
```

Em seguida o programa pedirá o número da linha inicial (entre 1 e 9900) e o incremento (1 a 9900). Para cancelar o comando, entre dois zeros quando o número de linha aparecer na tela. A rotina terminará com uma mensagem de erro, que deverá ser ignorada.

Para apagar um conjunto de linhas de um programa, digite:

```
RANDOMIZE USR 64000
```

O programa solicita que o usuário digite os números de linha do início e do fim do bloco a ser apagado. Para acionar o catalogador de fitas, chame:

```
RANDOMIZE USR 63919
```

A seguir, a borda da tela começará a piscar. Posicione a fita a catalogar no início de um programa e pressione a tecla **PLAY** do gravador. O programa exibirá na tela informações contidas no cabeçalho do arquivo (*header*).

Finalmente, para converter números decimais para hexadecimais, use:

```
RANDOMIZE USR 64394
```

Para efetuar a conversão no sentido oposto, digite:

```
RANDOMIZE USR 64453
```

Não é preciso pressionar <**ENTER**> após entrar o número hexadecimal.

```
 5 CLEAR 63488: BORDER 0:
 PAPER 0: INK 6: CLS
10 PRINT AT 0,5; INVERSE 1;"C
ONJUNTO DE FERRAMENTAS"
12 PRINT AT 8,2;" Qualquer te
cla para carregar  programa
em codigo de maquina.": PAUSE
0
14 LOAD "CREF"CODE
```

```
 15 CLS : PRINT "Pokeando codi
 go de maquina.      Prepare o
 cassete para gravar."
 20 LET L=100: RESTORE L: FOR
 N=63489 TO 64560 STEP 16
 30 LET T=0: FOR D=0 TO 15
 40 READ A: POKE (N+D),A: LET
 T=T+A: NEXT D
 50 READ A: IF A<>T THEN
 PRINT "ERRO DE CHECKSUM EM ";
 L: STOP
 60 LET L=L+10
 70 NEXT N
100 DATA 62,12,205,48,252,205,
60,250,237,67,155,248,42,83,92
,1,2019
110 DATA 0,0,126,254,128,40,9,
197,205,184,25,193,3,235,24,
242,1865
120 DATA 205,43,45,58,155,248,
205,40,45,239,4,56,205,162,45,
33,1788
130 DATA 15,39,167,237,66,48,2
,207,5,33,145,248,126,60,40,32
,1470
140 DATA 35,229,237,91,83,92,
42,75,92,167,237,82,68,77,235,
237,2079
150 DATA 177,197,229,245,204,
157,248,241,225,193,234,80,248
,225,24,220,3147
160 DATA 42,83,92,58,155,248,
54,0,35,119,205,40,45,239,49,
192,1656
170 DATA 56,42,83,92,205,184,
25,42,75,92,43,167,237,82,216,
235,1876
180 DATA 229,239,224,15,49,56,
205,162,45,225,112,35,113,43,
24,228,2004
190 DATA 201,224,228,235,236,
239,246,255,0,0,10,0,229,6,4,
35,2148
200 DATA 126,254,14,40,4,16,
248,225,201,197,35,35,35,78,35
,70,1613
210 DATA 42,83,92,217,1,1,0,
217,205,149,22,43,235,167,237,
66,1777
220 DATA 235,48,11,217,3,217,
197,205,184,25,193,235,24,234,
217,197,2442
230 DATA 217,209,42,155,248,
205,169,48,235,42,104,92,35,35
,115,35,1986
240 DATA 114,235,62,0,167,1,9,
0,237,66,56,17,60,1,90,0,1115
250 DATA 237,66,56,9,60,1,132,
3,237,66,56,1,60,209,225,229,
1647
260 DATA 245,130,214,4,245,6,0
,56,9,79,40,12,205,85,22,35,
1387
270 DATA 24,6,237,68,79,205,
232,25,193,241,197,79,6,0,9,65
,1666
280 DATA 229,197,35,35,235,42,
104,92,1,5,0,237,176,193,225,
229,2035
290 DATA 197,239,224,164,5,58,
193,164,4,224,1,3,225,192,2,56
,1951
300 DATA 205,213,45,193,225,
198,48,119,43,16,228,229,42,
104,92,35,2035
310 DATA 35,126,225,198,48,119
,241,193,245,42,83,92,167,229,
237,66,2346
320 DATA 225,48,8,197,205,184,
25,193,235,24,242,235,35,35,
293,126,2210
330 DATA 128,119,201,62,0,205,
48,252,205,26,31,33,0,0,62,0,
1372
340 DATA 237,66,229,193,205,43
,45,62,254,205,1,22,205,227,45
,201,2240
350 DATA 42,75,92,237,75,83,92
,62,0,237,66,229,62,1,205,48,
1606
360 DATA 252,193,205,43,45,205
,227,45,62,2,205,48,252,201,
221,33,2239
370 DATA 32,255,17,17,0,175,55
,205,86,5,62,3,205,48,252,221,
1638
380 DATA 33,32,255,221,126,0,
198,6,221,229,205,48,252,62,13
,215,2116
390 DATA 62,4,205,48,252,221,
225,221,35,221,126,0,254,255,
40,10,2179
400 DATA 6,10,221,126,0,221,35
,215,16,248,62,13,215,62,5,205
,1660
410 DATA 48,252,221,33,32,255,
221,78,11,221,70,12,195,163,
249,205,2266
420 DATA 142,250,62,10,205,48,
252,205,60,250,205,110,25,229,
62,13,2128
430 DATA 215,62,11,205,48,252,
205,60,250,205,110,25,193,32,
16,229,2118
440 DATA 35,35,126,35,95,126,
87,225,237,90,17,4,0,237,90,
229,1668
450 DATA 197,62,0,237,66,218,
87,252,195,89,252,6,0,197,205,
95,2158
460 DATA 252,205,115,252,193,
254,13,40,26,254,58,48,240,214
,48,56,2268
470 DATA 236,245,4,120,254,6,
32,4,5,241,24,225,241,245,198,
48,2128
480 DATA 215,24,218,221,33,49,
255,120,254,0,40,207,33,0,0,
221,1890
490 DATA 94,0,221,86,1,241,254
,0,40,7,237,90,56,13,61,32,1433
500 DATA 249,221,35,221,35,5,
32,231,229,193,201,207,5,42,75
,92,2073
510 DATA 237,75,83,92,237,66,
192,207,9,62,10,205,48,252,205
,60,2040
520 DATA 250,34,30,255,62,13,
215,62,12,205,48,252,205,60,
250,34,1987
530 DATA 28,255,33,48,48,34,59
,255,34,61,255,237,75,30,255,
205,1912
540 DATA 115,251,62,2,50,107,
92,50,107,92,205,149,23,205,
176,22,1708
550 DATA 62,0,205,1,22,33,59,
255,6,4,126,229,197,205,129,15
,1548
560 DATA 193,225,35,16,245,205
,44,15,205,23,27,221,33,58,92,
221,1858
570 DATA 203,0,126,32,13,42,89
,92,205,167,17,62,255,50,58,92
,1503
580 DATA 24,206,42,89,92,34,93
,92,205,251,25,120,177,32,10,
223,1715
590 DATA 254,13,40,174,205,176
,22,207,1,237,67,73,92,42,93,
92,1788
600 DATA 235,33,85,21,229,42,
97,92,55,237,82,229,96,105,205
,110,1953
610 DATA 25,32,6,205,184,25,
205,232,25,193,121,61,176,40,
47,197,1774
620 DATA 3,3,3,3,43,237,91,83,
92,213,205,85,22,225,34,83,
1425
630 DATA 92,193,197,19,42,97,
92,43,43,237,184,42,73,92,235,
193,1874
640 DATA 112,43,113,43,115,43,
114,237,75,28,255,205,115,251,
241,195,2185
650 DATA 195,250,33,62,255,126
,60,254,58,40,8,119,11,121,128
,176,1896
660 DATA 200,24,239,62,48,119,
43,24,236,62,14,205,48,252,205
,60,1841
670 DATA 250,62,13,215,197,62,
15,205,48,252,193,46,2,96,124,
203,1983
680 DATA 31,203,31,201,31,203,
31,230,15,205,189,251,215,124,
230,15,2207
690 DATA 205,189,251,215,97,45
,32,230,62,13,215,201,198,48,
254,58,2313
700 DATA 216,198,7,201,62,16,
205,48,252,17,85,255,6,4,213,
197,1982
710 DATA 205,95,252,205,115,
252,215,245,241,193,209,18,19,
16,239,62,2581
720 DATA 13,215,62,17,205,48,
252,221,33,85,255,17,0,16,33,0
,1472
730 DATA 0,14,4,221,126,0,221,
35,214,48,218,87,252,254,10,56
,1760
740 DATA 2,214,7,254,16,210,87
,252,71,254,0,40,3,25,16,1704
750 DATA 203,58,203,27,203,58,
203,27,203,58,203,27,203,58,
203,27,1964
760 DATA 13,32,208,229,193,205
,43,45,205,227,45,62,13,215,
201,203,2139
800 CLS : PRINT AT 5,5;" COMPI
LACAO COMPLETA. "
810 PRINT AT 7,2;"PREPARE O CA
SSETE PARA GRAVAR"
820 PRINT AT 9,4;"O NOME E TOO
LKIT"" CODE"
830 SAVE "TOOLKIT"CODE 63489,
2000
```

ROBÔS CONTROLADOS POR COMPUTADOR

Os computadores invadiram nossos lares e, em breve, será a vez dos robôs. Não se assuste: são apenas periféricos instrutivos, em nada semelhantes aos monstros mecânicos da ficção científica.

O cérebro humano comunica-se com o exterior através dos sentidos, para a entrada de dados, e através dos músculos, para a saída. Da mesma forma, um computador também pode ser ligado a canais de entrada e saída para se comunicar com o mundo exterior. Normalmente, computadores de uso geral, como os micros domésticos, estão ligados a uma série de dispositivos de entrada e saída — como o teclado, o vídeo, joysticks, impressora etc. Entretanto, é possível utilizá-los como "cérebro" para outros tipos de dispositivos eletrônicos e mecânicos, capazes de dotar o processador central de "sentidos", que recebem dados do mundo exterior, e de "músculos", que o fazem agir sobre esse mundo. Com isso, eles se transformam em robôs — ou, em outras palavras, dão "inteligência" a um dispositivo mecânico. Se conectado a uma máquina de escrever, por exemplo, o computador transforma-a em um eficiente processador de textos; se conectado a um órgão eletrônico, transforma-o em um poderoso sintetizador. Se você ligá-lo a um aspirador de pó, provavelmente terá um robô que limpa o carpete!

O QUE É UM ROBÔ?

Tantos robôs têm aparecido em filmes de ficção científica — como os famosos R2D2 e CP30, da série *Star Wars* — que pouca gente tem dúvidas sobre seu aspecto e suas funções. Alguns deles — os *andróides* — se assemelham aos seres humanos. Outros se parecem mais com um aspirador de pó. Em ambos os casos, porém, são "seres" mecânicos, dotados de habilidades manuais e de movimentação, sentido de visão, audição e tato (entre outros), e com um razoável grau de inteligência.

Os robôs reais ainda estão bem distantes do charme e inteligência dos robôs da ficção. A maioria deles não passa de braços mecânicos programáveis, que efetuam tarefas repetitivas e exatas, sem ter qualquer percepção do ambiente que os cerca. Se tirarmos um carro da linha de montagem de um robô industrial, por exemplo, ele continuará a soldar ou pintar "no vazio".

Já existem alguns robôs que possuem audição ou visão limitadas. Um robô de inspeção eletrônica, por exemplo, examina placas de circuitos impressos com uma câmara de vídeo, e é capaz de detectar falhas de montagem. Outro tipo de robô é capaz de apanhar uma peça em uma esteira rolante e reconhecê-la em qualquer posição que esteja. Outros, ainda, são capazes de pegar um ovo sem quebrá-lo, pois têm delicados sensores de pressão na extremidade de seus "dedos". Mas, entre os robôs existentes hoje, nenhum é dotado de "inteligência artificial" mais ampla, e não há, por enquanto, nenhuma função para robôs andróides.

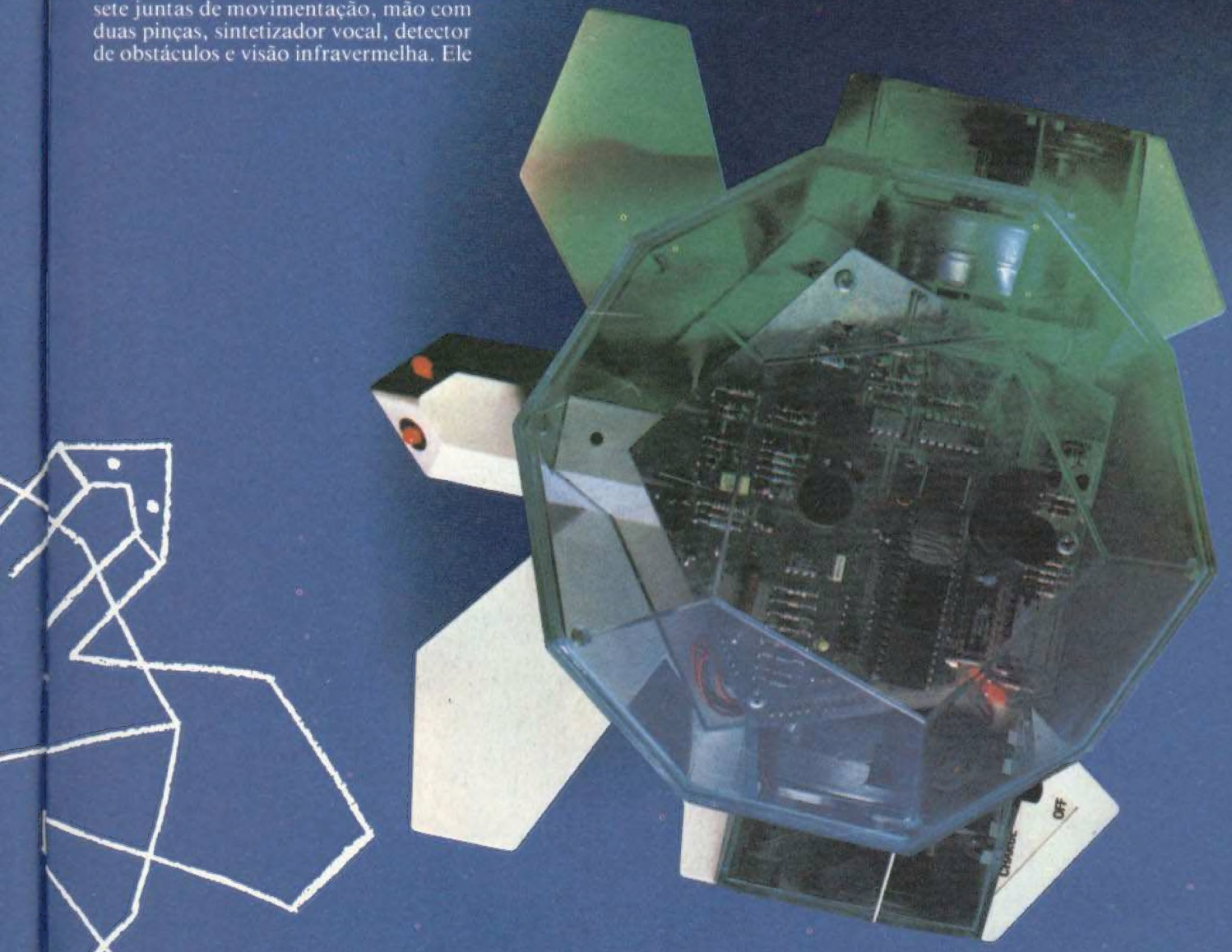
Muitos robôs são usados em atividades perigosas para o ser humano, como o manejo de materiais radiativos, tóxicos ou explosivos, ou em ambientes de temperatura ou pressão muito elevadas — para a inspeção de plataformas petrolíferas, por exemplo, já se recorre a robôs submarinos. Os robôs também

- INTELIGÊNCIA ARTIFICIAL
- COMUNICAÇÃO COM O MUNDO EXTERIOR
- CODIFICAÇÃO BINÁRIA
- TIPOS DE ROBÔ
- LINGUAGENS DE CONTROLE
- OUTROS RECURSOS
- SENTIDOS ROBÓTICOS
- A TARTARUGA

sao utilizados em tarefas monótonas e repetitivas, como solda, pintura, montagem e classificação de peças.

Embora ainda não existam "escravos domésticos" (aguardados ansiosamente pelas donas-de-casa), já foram lançados pequenos robôs móveis autônomos, como o Hero I, fabricado nos Estados Unidos, que tem rodas, um braço com sete juntas de movimentação, mão com duas pinças, sintetizador vocal, detector de obstáculos e visão infravermelha. Ele pode aprender a realizar tarefas simples mas úteis, como, por exemplo, vigiar uma casa à noite.

Portanto, um robô é apenas um conjunto de dispositivos mecânicos e eletrônicos, capaz de uma série limitada de percepções e movimentos. Esses dispositivos separados são, em geral, controlados por um processador digital, montado no interior do próprio robô, ou em um microcomputador de uso geral, situado fora dele.



COMUNICAÇÃO COM O EXTERIOR

Para se comunicar com um robô, os microcomputadores pessoais geralmente têm pelo menos uma porta de entrada/saída (E/S), através da qual podem entrar em contato com dispositivos mecânicos externos. Existem dois tipos de porta E/S: as seriais e as paralelas. Elas diferem entre si quanto à maneira de enviar e receber o fluxo de bits do periférico. Na maioria dos micros, a unidade básica de intercâmbio de informação é o byte — um conjunto de oito bits, ou dígitos binários. A informação armazenada em um byte é uma seqüência de oito algarismos 0 ou 1 — por exemplo: 00101101.

Na memória interna de trabalho do computador, dois bytes são rotulados especialmente para intercâmbio de dados com cada porta de entrada e saída. O primeiro byte, chamado de Registro de Direção dos Dados (*Data Direction Register*, DDR), determina o status da porta, definindo quando o intercâmbio será no sentido da entrada ou no sentido da saída. Em geral, se um bit no DDR tiver valor igual a 0, a porta correspondente está recebendo informações de fora; se for igual a 1, ela está transmitindo. Portanto, com o auxílio de um comando **POKE**, em BASIC, podemos ajustar cada um dos oito canais comandados pelo DDR, colocando um número decimal entre 0 (todos os canais recebendo) e 255 (todos os canais transmitindo, ou seja, DDR = 11111111 em binário). Colocando o valor 15, por exemplo, teremos DDR = 00001111 — isto é, os quatro primeiro canais estão emitindo, e os quatro últimos, recebendo. O endereço absoluto do DDR, a ser usado com o comando **POKE**, varia de computador para computador.

O segundo byte que controla a porta corresponde ao endereço da mesma. Um dado colocado nesse byte é convertido pela interface, conforme a aplicação a que se destina, em uma série de pulsos de voltagem, que são então passados para o dispositivo. No sentido oposto, as voltagens emitidas pelo dispositivo periférico são convertidas para bits pela interface, e colocadas no byte de endereço da porta. Dessa forma, o processador central — e qualquer programa introduzido nele — tem fácil acesso à porta de entrada e saída, não precisando se ocupar com as conversões a serem realizadas.

A linguagem de máquina, como algumas linguagens de alto nível, entre elas o BASIC, dispõe de instruções para escrever e ler numa determinada porta de saída. Essas instruções são denominadas **OUT** e **INP**, respectivamente. Em BASIC, por exemplo, um comando:

```
OUT 32,127
```

coloca o número decimal 127 na porta de saída número 32. Normalmente, a maioria dos micros de oito bits admite até 256 portas de E/S, numeradas de 0 a 255. Embora algumas delas estejam reservadas para periféricos já existentes, como o gravador cassete, as outras estão disponíveis para outros usos, como o controle de um robô.

O comando **INP** tem a mesma sintaxe e é usado para ler um byte no endereço da porta. Em geral, o computador não sabe quando este dado está disponível. Assim, costuma-se usar um outro comando em BASIC para testar a disponibilidade de um novo dado: **WAIT** (não confundir com o comando **PAUSE** dos micros da linha Sinclair).

Em uma porta paralela, os oito bits disponíveis no periférico ou byte de endereço da porta são transmitidos, simultaneamente, por oito fios distintos. Na porta serial, a transmissão se realiza em um bit de cada vez, em "fila indiana", por um único fio.

Quando chega a seu destino, no periférico, a informação é usada para controlar várias operações. Em um dispositivo eletromecânico, como um robô, ela é utilizada para ligar e desligar motores, relés etc. Após selecionar o endereço da porta de saída, o programador pode enviar sinais para destinos específicos. Por exemplo, se o valor é ajustado em 00110111, um sinal é mandado para as destinações correspondentes, através do terceiro, quarto, sexto, sétimo e oitavo condutores de uma porta paralela, ou, seqüencialmente, por um fio só, de uma porta serial.

ROBÔS PARA MICROCOMPUTADORES

Com o aumento do interesse pela robótica, surgiram nos últimos anos vários tipos de robôs baratos que podem ser controlados por um computador doméstico. As duas categorias básicas de robôs desse tipo são os braços robóticos e os robôs móveis (tartarugas e *buggies*), geralmente fornecidos em forma de kits para montar, e com o software necessário para operá-los.

Os braços robóticos procuram imitar os movimentos do braço humano. Costumam ter cinco pontos de movimento, ou *graus de liberdade*: ombro, cotovelo, pulso, garra (com ou sem rotação ao redor do pulso) e base do braço (movimento basculante). Braços robóticos podem ter até nove graus de liberdade. O nível de sofisticação do braço se reflete, é claro, em seu preço final.

Os braços mais baratos, como o Armatron, comercializado na Europa, não são programáveis. A designação de robô não se aplica a eles, portanto — não passam de um brinquedo mais elaborado. Outros, como o Armbot e o Armdroid 1, podem ser controlados por vários tipos de microcomputador. Têm cinco ou seis motores de passo variável e custam tanto quanto uma UCP de bom preço. Seus movimentos, realizados um por vez ou de modo contínuo, são controlados de modo direto ou por uma seqüência armazenada no micro. É possível adicionar pausas, alterar a velocidade do braço e editar as seqüências armazenadas, para modificar ou acrescentar novas fases de movimento. O programa para controle do braço pode ser escrito em Assembler ou BASIC, com chamadas às rotinas em código de máquina que controlam as portas de E/S. Estas são fornecidas pelo fabricante que também coloca à disposição do usuário listagens-exemplo de BASIC, com comentários e explicações.

Existem ainda braços robóticos que utilizam servomotores, em vez de motores de passo. Um servomotor possui um sensor de posição que verifica continuamente se as posições determinadas por comandos estão sendo atingidas. Circuitos integrados de preço bastante acessível (amplificadores lineares) permitem um controle razoável do periférico. Um exemplo de braços desse tipo é o Beasty, comercializado na Inglaterra para os micros da linha BBC.

O Beasty tem diversos implementos úteis para o desenvolvimento de projetos de robótica aplicada. Uma câmara miniaturizada, chamada Snap, permite que o micro "veja" e mova o braço. Esse dispositivo contém um circuito integrado sensível à luz, que transmite uma imagem digitalizada com uma resolução de 128 × 256. Pesa menos que 45 g, mede 8 × 10 cm, e é capaz de captar até vinte imagens por segundo. Vários programas interessantes são fornecidos com o Snap, possibilitando, entre outras coisas:

- exibir as imagens captadas no vídeo do microcomputador. A imagem pode ser "fixada", como se fosse uma foto, armazenada em disco ou fita, e, posteriormente, reproduzida por meio de uma impressora gráfica;

- animar uma seqüência de imagens (vinte quadros), como se fosse um filme ou desenho animado;

- comparar duas imagens tomadas em períodos distintos. Esse recurso tem larga aplicação, por exemplo, na monitorização de residências, edifícios ou lojas, no sentido de evitar a penetração de estranhos. Se a imagem tiver se modificado além de um limite prefixado, um alarme pode ser ativado. O programa mostra um gráfico com o número de mudanças, em função do tempo;

- reconhecer cenas e objetos e movimentar o braço em função disso.

Muitas empresas que fabricam jogos mecânicos de armar, como Meccano e Lego, lançaram kits com braços robóticos, que podem ser conectados a microcomputadores baratos, como o Sinclair Spectrum. Esses jogos são muito educativos — aprende-se muito sobre o funcionamento de um robô, quando se tem a oportunidade de armar e programar um sistema desde o início.

ROBÔS MÓVEIS

Ao contrário dos braços robóticos, de base estacionária, alguns robôs para microcomputadores, como as tartarugas e os *buggies*, movimentam-se sobre rodas. O primeiro robô desse tipo data dos anos 40, quando o cientista britânico Ross Ashby desenvolveu uma tartaruga eletrônica que tinha a tarefa de achar uma tomada para alimentar suas baterias. O "bichinho", apesar de muito simples, exibia notáveis e surpreendentes padrões "comportamentais".

A primeira tartaruga para micros apareceu no Laboratório de Inteligência Artificial do MIT (*Massachusetts Institute of Technology*), nos Estados Unidos. Era o "filhote" da equipe do Prof. Seymour Papert, criador da linguagem LOGO. Sua estrutura também é simples: consta de uma campânula hemisférica de acrílico, montada sobre uma base móvel com duas rodas independentes, acionadas por motores controlados pelo microcomputador. A campânula pode ser acoplada a um sensor de contato, que se comunica com o computador quando o robô colide com algum objeto ou parede. Na "barriga" da tartaruga, um solenóide controla um porta-caneta, que levanta ou abaixa sob controle do computador. Assim, a tartaruga pode ser "ensinada" a traçar desenhos sobre um papel colocado no chão.

As tartarugas têm sido muito utilizadas em escolas secundárias, para ensinar conceitos de programação a crianças menores. Empregando a linguagem LOGO ou uma simplificação da mesma (com comandos de uma tecla), a criança aprende a programar a tartaruga para realizar movimentos complexos. Um dos jogos preferidos consiste em dar comandos pelo teclado, até que a tartaruga entre em sua "casinha". Os comandos em LOGO, digitados em um microcomputador, são traduzidos por uma interface de software para códigos binários de controle das portas de saída. Esses sinais são enviados por um "cordão umbilical" ligado à tartaruga, ou, nos modelos mais sofisticados, como a *Valiant Turtle*, por meio de sinais infravermelhos. Uma placa com circuitos eletrônicos, na tartaruga, traduz os comandos binários em ações sobre os motores das rodas e da caneta. A tartaruga tem ainda um olho luminoso, que acende quando ela se movimenta.

Os comandos do LOGO permitem a movimentação da tartaruga segundo um sistema geométrico, em que a própria tartaruga é o ponto de referência. Examinaremos esses comandos em detalhe, na série de artigos sobre novas linguagens. Adiantamos aqui, entretanto, as instruções mais típicas:

FORWARD - Move a tartaruga, em linha reta, para a frente.
BACK - Move a tartaruga, em linha reta, para trás.
RIGHT - Vira a tartaruga para o lado direito.
LEFT - Vira a tartaruga para o lado esquerdo.
PENUP - Levanta a caneta, interrompendo o desenho.
PENDOWN - Abaixa a caneta, permitindo o desenho.
REPEAT - Repete uma seqüência de movimentos várias vezes.
TO...END - Armazena uma seqüência programada de movimentos, que recebe um nome (procedimento).

Para fazer a tartaruga desenhar um triângulo com lados de duzentos passos (unidades de deslocamento), as instruções a serem dadas são:

```
FORWARD 200
RIGHT 120
FORWARD 200
RIGHT 120
FORWARD 200
RIGHT 120
```

Poderíamos escrever o mesmo com mais economia:

```
REPEAT 3 [FORWARD 200 RIGHT
120]
```

Ou, ainda, armazenar as instruções na forma de um procedimento denominado **TRIANGULO**:

Há robôs para micros no Brasil?

Os leitores que se interessaram pelo assunto com certeza ficarão frustrados com a resposta: atualmente não há, no Brasil, robôs comerciais para conexão a microcomputadores.

A Universidade Estadual de Campinas (Unicamp) construiu uma tartaruga semelhante à criada pelo MIT, para uso em seus projetos com a linguagem LOGO. Porém, essa tartaruga não chegou a ser produzida industrialmente. Grupos de trabalho da Universidade de São Paulo (USP) e da Universidade Federal do Rio de Janeiro (UFRJ) desenvolveram protótipos de braços robóticos, como os descritos no artigo, mas estes também não são comercializados ainda.

Para quem tem condições, resta a possibilidade de mandar trazer do exterior um dos modelos de robô mais conhecidos, como a *Terrapin Turtle* ou a *Valiant Turtle* (tartarugas), o Hero I (robô *buggy* da Zenith/Heathkit), Armbot, Armdroid ou Beasty (braços robóticos) e kits da Lego.

```
TO TRIANGULO
REPEAT 3 [FORWARD 200 RIGHT
120]
END
```

Uma vez definida, a palavra **TRIANGULO** passa a fazer parte do vocabulário LOGO. Depois, para traçar a figura, basta digitar o comando **TRIANGULO** pelo teclado.

Pode-se utilizar um mesmo procedimento dentro de outros, destinados a realizar movimentos mais complexos. Por exemplo, para fazer com que a tartaruga desenhe uma flor formada de doze pétalas triangulares iguais, digitamos o seguinte:

```
TO FLOR
REPEAT 12 [TRIANGULO RIGHT 30]
END
```

O LOGO é uma linguagem muito versátil, também empregada para fazer cálculos matemáticos, manipular dados simbólicos como listas, palavras e nomes etc. Associado a uma tartaruga, tem o poder de tornar muito divertido o aprendizado da programação.

Programas em outras linguagens, como o BASIC, também podem ser usados para controlar a tartaruga.

A TORRE DE BABEL

- RAÍZES DE UMA LINGUAGEM
- EVOLUÇÃO
- LINGUAGENS IMPERATIVAS
- LINGUAGENS FUNCIONAIS
- QUARTA GERAÇÃO

Existem atualmente mais de duzentas linguagens de programação. Algumas delas prometem tornar-se tão conhecidas quanto o BASIC no decorrer dos próximos anos.

Até agora, focalizamos em *INPUT* duas linguagens de programação: Assembler e BASIC. Estas são, de fato, as linguagens mais difundidas entre os usuários dos microcomputadores pessoais e domésticos existentes atualmente no mercado brasileiro. Não são, porém, as únicas disponíveis.

Algumas linguagens vêm se tornando gradativamente mais conhecidas e utilizadas em micros pessoais — em particular o LOGO, o PASCAL e o FORTH. Outras, criadas já há algum tempo, mas destinadas a computadores de maior porte, prometem estender seu uso aos micros, podendo se tornar o futuro "esperanto" das máquinas de menor porte. Entre elas, incluem-se o LISP e o PROLOG. Entretanto, como as linguagens mais recentes ainda se encontram em processo de aceitação e difusão, nem sempre estão disponíveis para os micros de todas as marcas.

Na série que se inicia com este artigo, apresentaremos duas linguagens de programação cuja popularidade tem crescido dia a dia entre os usuários de micros: o LOGO e o PASCAL. Cada uma tem características que permitem sua aplicação seja como linguagem geral de programação (como o PASCAL), seja como linguagem destinada a áreas específicas — a área educativa, por exemplo (como o LOGO).

Examinaremos aqui alguns aspectos do desenvolvimento de linguagens de programação, a evolução histórica das linguagens existentes e os critérios usados em sua classificação.

Há atualmente mais de duzentas linguagens de programação catalogadas. À primeira vista, podemos ter a impressão de que se trata de uma enorme e bíblica "torre de Babel". No entanto, as linguagens de programação, tal como as linguagens naturais, podem ser enquadradas em linhas evolutivas, ou *famílias* de linguagens. Dentro de uma família, as semelhanças são maiores que as diferenças, o que facilita muito a compreensão e o aprendizado das linguagens que a compõem.

NOVAS LINGUAGENS

Um dos fatores condicionantes do aparecimento de novas linguagens de programação é o progresso tecnológico do hardware dos computadores — ou seja, a disponibilidade de memórias centrais cada vez maiores, o aumento na velocidade do processador central (UCP), o custo mais baixo das memórias auxiliares de disco, o aparecimento de redes de computadores etc.

Nos últimos anos, as linguagens de alto nível — que se parecem mais com a linguagem natural humana do que com a binária, entendida somente pelos computadores — sofreram grande evolução. Avanços técnicos recentes permitem que elas sejam utilizadas como a linguagem nativa dos computadores, no lugar da linguagem de máquina. Com isso, a programação se torna muito mais fácil, podendo ser realizada, em muitos casos, pelos próprios usuários.

No final da década de 70, o Assembler e o BASIC eram praticamente as únicas linguagens para micros de custo mais baixo, pois o tamanho das memórias ROM disponíveis limitavam muito a complexidade e a extensão do programa interpretador, que deveria ficar permanentemente na memória da máquina.

Entretanto, à medida que o espaço para a memória foi sendo ampliado, cresceu também a demanda por outros tipos de linguagem — o que era de se esperar, dado o número cada vez maior de usuários e de possibilidades de aplicação. Surgiram então as primeiras implementações práticas, em microcomputadores, de FORTH, PASCAL, LOGO, FORTRAN, PILOT, C, LISP, PROLOG, SMALLTALK e várias outras linguagens. Muitas delas só podiam ser usadas, anteriormente, em computadores de grande porte.

Mesmo algumas linguagens de maior complexidade, que encontram aplicações em áreas comerciais ou científicas mais "pesadas", como o COBOL, APL, PL/1, SNOBOL, MODULA, ALGOL e ADA, já podem ser utilizadas em microcomputadores da faixa profissional.

Ao que tudo indica, a difusão dessas linguagens tende a crescer, conforme os usuários forem se familiarizando com as suas vantagens. A expansão na capacidade e velocidade dos micros fornecerá as bases para esse avanço.

Outro fenômeno registrado na microinformática é a multiplicação de programas que funcionam,ao mesmo tempo, como um aplicativo (bancos de dados, planilhas eletrônicas etc.) e como linguagem de programação. É o caso do dBASE II (banco de dados) e do FRED (planilha eletrônica), entre outros.

O desenvolvimento de muitos desses novos aplicativos "genéricos" deu origem a linguagens altamente específicas, usadas para simulação (DYNAMO, GPSS), controle de equipamentos industriais (PEARL), controle de interfaces e equipamentos musicais (AMPLE) etc.

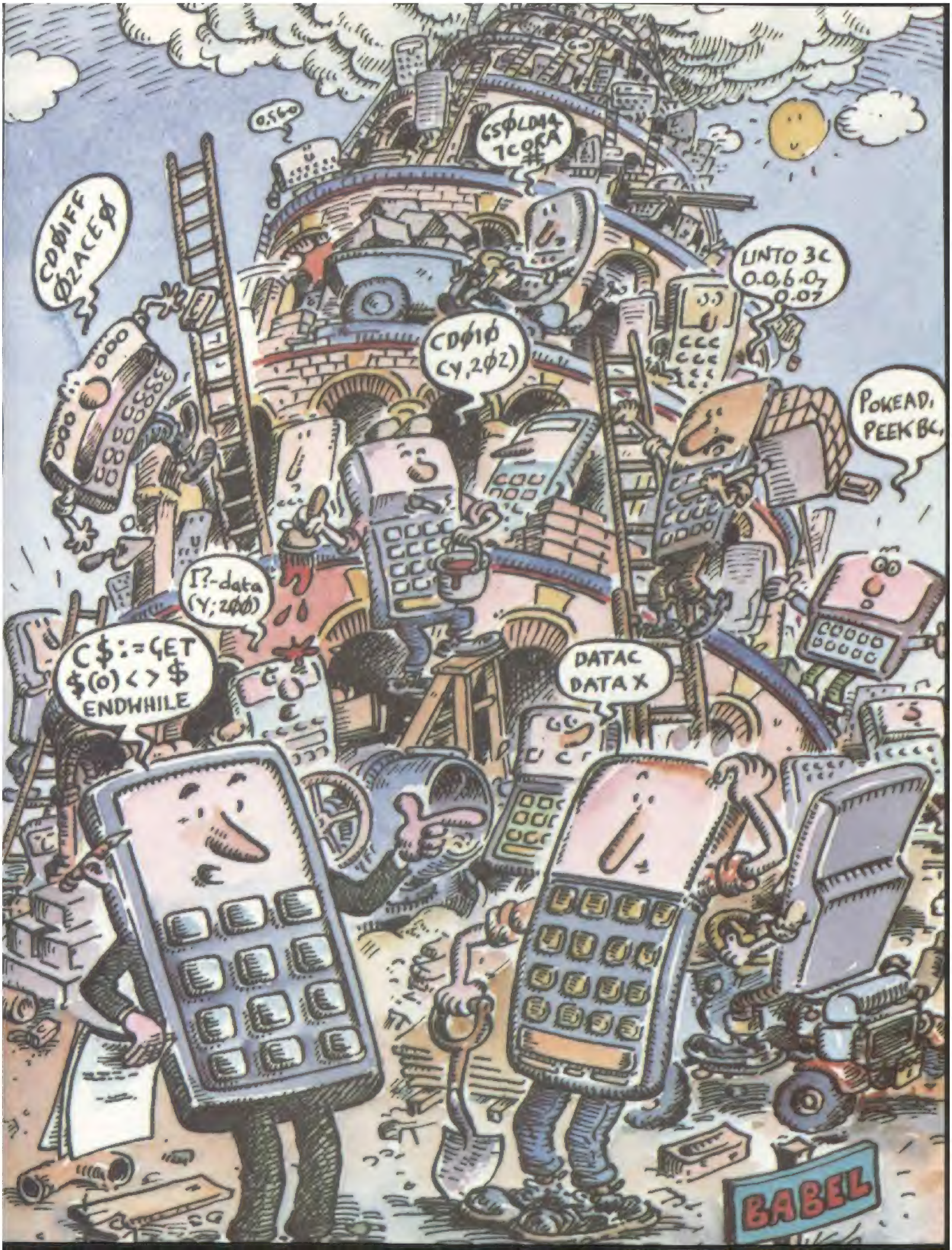
O DESENVOLVIMENTO DE LINGUAGENS

Nada impede que todas as linguagens de programação venham a ser desenvolvidas para microcomputadores. Mas parece pouco provável que isso ocorra, visto o grande número de linguagens existentes. Das mais de duzentas desenvolvidas até agora, não chegam a vinte as que se tornaram conhecidas e utilizadas.

Como as linguagens naturais, as linguagens de programação passam por contínuo processo de transformação, revisão e alteração. O FORTRAN (*FORmula TRANslation*), por exemplo, evoluiu tanto a partir da versão original, lançada em 1954, que, em alguns aspectos, é uma linguagem diversa.

Além desse contínuo aperfeiçoamento, as linguagens também passam por um processo de geração de uma enorme quantidade de "dialetos" ou variantes, desenvolvidas para implementação em linhas específicas de hardware, em diferentes sistemas e organizações.

Uma das limitações mais sérias encontradas pelo usuário de micros é justamente a incompatibilidade entre diferentes dialetos do BASIC. Um progra-


CDØIFF
Ø2ACEØ
CDØ1Ø
(Y,2Ø2)
UNTO 3C
0.0,6.07
0.07
POKEAD,
PEEK BC,
I?-data
(Y;2ØØ)
C$:=GET
$(0)<>$
ENDWHILE
DATAC
DATA X
BABEL

ma desenvolvido segundo o BASIC da linha Apple, por exemplo, em geral não pode ser executado em um micro de outra linha, como o Spectrum ou o MSX.

Apesar de todo o esforço despendido na elaboração de um padrão internacional, o ANSI BASIC, o problema ainda não foi solucionado. A inadequação desse padrão é evidente: tanto que não há uma única marca de computador que o adote integralmente. Por isso, o interpretador BASIC, criado pela firma norte-americana Microsoft, tem sido usado como uma espécie de padrão por fabricantes de máquinas tão distintas como o Apple, o TRS-80 e o MSX.

O projeto do MSX, aliás, é fruto de uma das últimas tentativas de se padronizar hardware e software de microcomputadores pessoais através da implementação de mais um dialeto do BASIC Microsoft. Esse padrão tem tido excelente repercussão mundial, principalmente no Japão (seu país de origem), na Europa, e no Brasil.

A história das outras linguagens não foi muito diferente: o sucesso de um padrão resultou sempre da conquista de uma boa fatia do mercado pela empresa que o desenvolveu. O FORTRAN, por exemplo, foi desenvolvido pela IBM e, como era distribuído gratuitamente com os computadores dessa empresa, passou a ser amplamente adotado e seguido, inclusive por outros fabricantes. Contando com uma grande base de usuários, a própria IBM pôde realizar melhorias subseqüentes do FORTRAN — como o FORTRAN II, o FORTRAN IV, as versões G e H, o FORTRAN 77 etc.

Entretanto, uma linguagem como o PASCAL, que talvez seja a segunda linguagem mais popular para micros, possui pelo menos sete versões diferentes. Cada uma delas é um semipadrão para o PASCAL relativo ao sistema operacional sob o qual roda.

AS RAÍZES DA LINGUAGEM

A torre de Babel é uma conseqüência inevitável da necessidade de desenvolvimento de novas linguagens.

Voltemos ao exemplo do FORTRAN. Seguramente, foi a primeira linguagem de alto nível desenvolvida. Sua criação atendeu a duas exigências: a resolução de fórmulas matemáticas e científicas, e o uso eficiente dos recursos computacionais disponíveis. Nesses aspectos, foi um sucesso. Mas, em dois outros pontos, que não tinham sido julgados importantes pelos projetistas do FORTRAN, a linguagem mostrou-se deficiente. Ela é excelente para fins matemáticos e científicos, mas extremamente inadequada (ou mesmo inútil) para tarefas como processamento de textos ou bancos de dados. Além disso, não é uma linguagem estruturada, o que restringe a ação do programador.

Na década de 60, desenvolveu-se uma outra linguagem, o ALGOL 60, com uma filosofia distinta de projeto. Seus pontos fracos e fortes diferem bastante daqueles que caracterizam o FORTRAN — sobretudo por se tratar da primeira linguagem de programação estruturada. Ela foi projetada para ser essencialmente algorítmica (daí seu nome, *ALGOrithmic Language*), ou seja, para corresponder não às fórmulas matemáticas, mas ao processo inerente usado na resolução de um problema (algoritmo). A unidade básica do ALGOL é o *procedimento*, um bloco autônomo de código, com uma tarefa claramente definida, que trabalha com tipos predefinidos de variáveis (*declarações*). Por essa razão, todas as linguagens derivadas diretamente do ALGOL, como o PASCAL, o MODULA-2 e o ADA, são classificadas como *procedimentais, declarativas, algorítmicas* e *estruturadas*.

Embora não tenha sido muito usado, o ALGOL 60 desempenhou um papel de destaque, gerando um grande número de linguagens. Já o COBOL, considerado o terceiro "avô" da computação, foi intensamente utilizado — mas só deixou um "descendente", o PL/1.

O COBOL (*Common Business — Oriented Language*) surgiu como resposta à conclusão — enunciada pelo Ministério da Defesa dos Estados Unidos — de que as instalações militares e centros de pesquisa necessitavam de uma linguagem de programação comum, se quisessem operar o sistema americano de defesa como um todo, efetivamente. Desenvolvida a partir de um esquema de cooperação entre governo e indústria, essa linguagem difundiu-se rápida e amplamente. É provável que seja, hoje, a linguagem mais utilizada para aplicações comerciais em computadores de grande porte. Uma das razões que explicam a grande aceitação do COBOL é sua proximidade do inglês natural.

EVOLUÇÃO

As linguagens de alto nível anteriormente mencionadas deram origem à grande maioria das linguagens de programação da década de 60, muitas das quais ainda em uso. O BASIC, por exemplo, é um descendente direto do FORTRAN. As diferenças entre ambos refletem o intuito de se chegar a uma linguagem de fácil uso pelos iniciantes em programação. O ALGOL 60 deu origem ao ALGOL 68 — tão diferente, que pode ser considerado uma nova linguagem, apesar do mesmo nome —, bem como ao PASCAL e ao SIMULA. De uma combinação entre o ALGOL e o COBOL surgiu o PL/1. Do PASCAL e do SIMULA nasceu um grupo de linguagens procedimentais e algorítmicas, que incluem o MODULA, o MESA, o EUCLID e o ADA. Finalmente, uma linguagem bastante poderosa, voltada à programação de objetos, originou-se do SIMULA — o SMALLTALK.

LINGUAGENS IMPERATIVAS

Todas as linguagens citadas são classificadas como *imperativas*, pois operam através de uma seqüência de comandos declarativos — isto é, comandos que determinam um tipo altamente específico de ação sobre dados da memória, rotulados na forma de variáveis. Muitas pessoas estão tão acostumadas a essa forma de programação, que imaginam tratar-se da única possível. Entretanto, outras famílias, que incluem linguagens como o FORTH, o LISP e o LOGO, têm estruturas de dados e de processamento bastante diferentes. Elas constituem o grupo de linguagens chamadas *funcionais* ou *aplicativas*, operando basicamente através da aplicação recursiva de funções.

LINGUAGENS FUNCIONAIS

As linguagens funcionais, como o próprio nome sugere, atuam sobre os dados por meio de funções ou procedimentos. Um procedimento é algo bem mais poderoso do que um comando. Vejamos por quê: toda linguagem de programação tem um vocabulário fixo de instruções, chamado *conjunto de instruções*. As linguagens imperativas utilizam um conjunto *fixo* de instruções, ou seja, um conjunto que não pode ser ampliado pelo próprio usuário. A linguagem funcional, ao contrário, não faz distinção entre os comandos, declarações ou funções pertencentes ao conjunto original e as funções ou procedimentos criados pelo usuário. Por isso, diz-se que certas classes de linguagens funcionais são *extensíveis*.

O processamento baseado em funções apresenta ainda outra vantagem: ela age sobre parâmetros de entrada e devolve parâmetros de saída, sem "saber" necessariamente o que eles representam. Isso justifica a classificação das funções como *generalizáveis*.

As linguagens funcionais podem ser divididas em módulos (blocos autônomos de processamento) mais facilmente do que as imperativas, e possuem uma estrutura interna hierárquica que é usada tanto para o armazenamento do programa quanto dos dados. No LISP ou no LOGO, por exemplo, uma lista definida como uma estrutura de dados pode ser usada para armazenar um programa. Existem comandos para modificar e executar um programa, armazenado em uma lista (**EXECUTE**, em LOGO, e **EVAL**, em LISP). Essa propriedade de uma linguagem é chamada *indireção*.

Já em uma linguagem imperativa, como o BASIC, programa e dados são elementos inteiramente distintos. O conjunto de instruções não pode ser ampliado e a ordem de execução é muito importante: se ela sofrer alguma modificação, determinados valores de memória não serão criados ou atualizados.

Linguagens como o LISP permitem uma abordagem bem diferente. Em um certo sentido, cada programa em LISP — ou nas linguagens que dele derivam — é uma extensão da linguagem original, com o acréscimo das novas funções definidas pelo usuário. Com uma linguagem funcional, o programador tem maior controle sobre o que faz do que com uma linguagem imperativa.

As três propriedades mencionadas — recursão, extensibilidade e indireção — são fundamentais em um campo muito atual das ciências da computação: a Inteligência Artificial ("raciocínio" e "aprendizado" por máquina). LISP e LOGO, aliás, foram criados especificamente para facilitar o processamento simbólico, não-numérico, característico das aplicações de Inteligência Artificial. Ambos nasceram no Laboratório de Inteligência Artificial do MIT (Massachusetts Institute of Technology, nos EUA).

EXTENSÕES

Como já dissemos, as linguagens de computador, da mesma forma que as linguagens naturais, não permanecem estáticas: elas evoluem constantemente, absorvendo aspectos de outras linguagens. Com isso, as diferenças entre linguagens imperativas e linguagens funcionais tendem a diminuir.

Tomemos como exemplo o BASIC: os dialetos mais difundidos do Microsoft BASIC (para as linhas Apple, TRS-80, TRS-Color e MSX) são linguagens puramente imperativas. Seus descendentes mais modernos, porém, como o QuickBASIC (da própria Microsoft), possuem características funcionais, entre elas a programação estruturada.

Um processo semelhante ocorreu com o BASIC original das linhas Sinclair. O SuperBASIC do modelo QL Spectrum, que dele deriva, tem não só estruturas — como **CASE SELECT** — próprias do PASCAL, como também oferece a possibilidade de se chamar funções apenas pelo nome e de se utilizar a recursão. Desenvolvimentos futuros da linguagem BASIC provavelmente a afastarão ainda mais do padrão original (ANSI), incorporando elementos de linguagens funcionais como o FORTH e o LISP.

Como já vimos, o LISP foi desenvolvido como uma linguagem para o campo de Inteligência Artificial, e, ainda hoje, predomina nessa área. Uma de suas principais características é o intenso uso de cadeias de caracteres (*string*). Outra linguagem útil na mesma área é o SNOBOL (*StriNg Oriented SymBOlic Language*), originalmente desenvolvida para o processamento de linguagem natural em Inteligência Artificial, Lingüística etc., em tarefas como a determinação da autoria de manuscritos, análise estilística de textos literários e geração automática de poemas.

Também merece destaque a recente evolução das *linguagens de desenvolvimento de sistemas*. São linguagens de alto nível, mas que incorporam recursos avançados de trabalho direto com a memória, processador e periféricos do computador. Entre as linguagens desse tipo, a de maior sucesso é o C, que foi usada para desenvolver o sistema operacional UNIX. Ela possui um poderoso conjunto de ferramentas de desenvolvimento de software e tem sido utilizada sobretudo por programadores mais experientes. Para o usuário comum, o C parece um tanto inacessível — está entre uma linguagem de alto nível, como o PASCAL, e um Assembler, bem mais flexível, mas difícil de usar. Mesmo assim, essa linguagem vem se difundindo, principalmente entre os amadores que desejam dominar sua máquina sem ter o trabalho de aprender Assembler. Existem diversas implementações do C para micros, inclusive o Spectrum.

O SmallTalk é outra linguagem que segue de perto a filosofia do LISP. Típica de uma nova família, a das *linguagens de programação de objetos*, ela constitui um "ambiente" completo de programação, como o C, e não apenas uma linguagem. O SmallTalk tem, no entanto, diversos recursos voltados para a máxima simplificação do conceito de programa e da tarefa de programação. Nessa linguagem, cada item de um programa é considerado um objeto. A programação é feita através do intercâmbio de "mensagens" entre objetos. Como não existe outro tipo de processamento, qualquer ação em um programa é sempre a mesma coisa: uma mensagem enviada para um objeto, que, por sua vez, envia mensagens para outros objetos.

Os objetos podem realizar processamentos internos, conforme a mensagem que recebem. Um objeto do SmallTalk é, assim, semelhante a uma função LISP ou a um procedimento PASCAL. Só que, em vez de construir procedimentos e incorporá-los a uma estrutura em árvore, como no LISP e LOGO, os objetos podem ser considerados entidades independentes — como os registros de uma base de dados, em que cada registro interroga ou responde aos demais. Em outras palavras, um objeto pode conter tanto dados quanto programas.

A idéia não é muito fácil de entender, a não ser que se observe o funcionamento do sistema. Infelizmente, ainda não se encontram implementações de SmallTalk para todos os micros. Mas certamente vale a pena investigar essa linguagem, pois ela representa uma tendência importante para o futuro. Sistemas operacionais de micros modernos, como o MacIntosh, da Apple, foram muito influenciados pela filosofia SmallTalk, inclusive quanto ao tratamento da interface com o usuário, que é feita através de "janelas" múltiplas, menus e acionamento do "mouse".

LINGUAGENS DE QUARTA GERAÇÃO

Certas linguagens aplicativas mais recentes (de quarta geração) foram desenvolvidas a fim de dotar o programador de poderosíssimos recursos, denominados *macros*. Com efeito, nessas linguagens, um único comando ou função equivale a centenas ou milhares de instruções de uma linguagem imperativa comum, como o BASIC. Um exemplo de linguagem de quarta geração para micros, o dBASE II (e sua extensão para micros de dezesseis bits, o dBASE III), é, ao mesmo tempo, um sistema aplicativo — mais especificamente, um sistema gerenciador de bases de dados, ou SGBD — e uma linguagem de programação.

Outra linguagem de quarta geração em franca ascensão é o PROLOG (*PROgramming in LOGic*), desenvolvido na França e adotado pelos japoneses em seu programa de computadores de quinta geração. O PROLOG está para o campo da Inteligência Artificial como o dBASE para a área de gerenciamento de dados. Existem PROLOG para micros de oito e dezesseis bits.

AVALANCHE: LISTAGEM COMPLETA

O jogo já foi todo digitado. Willie está pronto para escalar a montanha, enfrentar o mar, as pedras, as cobras e os buracos. Será que o programa vai funcionar?

Avalanche agora está completo, mas pode precisar de alguns acertos. É pouco provável que você tenha digitado um programa tão longo sem cometer alguns enganos. Dedique-se à resolução de eventuais problemas causados ao criar ou copiar erros. Para descobri-los, verifique suas listagens Assembly; se achar conveniente, remonte-as.

Depois que as rotinas separadas estiverem funcionando, carregue-as na memória, salve-as numa fita e teste o conjunto completo, chamando-o com o comando de execução de código de máquina de seu microcomputador.

Por sua estrutura modular, o programa do videogame *Avalanche* é fácil de corrigir. Caso encontre algum problema durante a execução do jogo, tenha o cuidado de identificar a parte que apresentou o defeito e concentre nela seus esforços para descobrir o erro.

Mesmo que todas as rotinas funcionem adequadamente quando rodadas uma a uma, é possível que o programa completo apresente falhas. Rotinas chamam outras rotinas — e se uma delas contiver algum tipo de erro ou estiver montada em lugar inadequado, o programa poderá não funcionar. Estamos, por isso, publicando uma listagem hexadecimal completa do jogo. Utilize-a para checar sua versão final.

Depois de montar *Avalanche*, o programa estará na memória do computador como uma série de números hexadecimais em posições sucessivas, tal qual mostramos neste artigo. O número de quatro dígitos na coluna esquerda corresponde ao endereço da posição de memória ocupado pelos dois primeiros bytes de programa naquela linha. Os pares subseqüentes de dígitos ocupam as posições seguintes. Você tem, portanto, a possibilidade de examinar cada byte de *Avalanche* através do conteúdo da posição de memória correspondente e verificar se cada instrução do jogo foi montada corretamente.

Ao contrário de alguns outros jogos de computador, *Avalanche* nunca chega a entediar, pois seus parâmetros de execução podem ser alterados. A construção modular do programa deixa o usuário livre para fazer as modificações que quiser, permitindo-lhe adaptar *Avalanche* às suas preferências pessoais. Seja curioso. Experimente. Afinal, o jogo agora é seu. E você pode fazer dele o que bem entender.

S

O jogo está escrito na região 50000 da memória para que possa ser montado pelo Assembler de *INPUT*.

Se você não está usando o Assembler de *INPUT*, talvez precise deslocar o jogo na memória por causa do espaço ocupado pelo próprio Assembler. A melhor alternativa é a região 20000. Para fazer a transferência basta mudar para 2 os endereços que iniciam com 5.

Lembre-se de verificar se a rotina que executa a música não apagou outras rotinas ao ser deslocada.

Use o monitor de código de máquina para analisar a memória do seu micro e ver se ela confere com a listagem em hexa dada a seguir. Se você tiver deslocado o jogo, as posições de memória serão diferentes, mas a seqüência de códigos hexa deve ser a mesma.

```
DEA8 18 3C 3C 18 3C 3C 3C 3C
DEB0 3C 3C 18 18 18 18 18 1E
DEB8 01 03 03 01 00 01 01 01
DEC0 80 C0 60 80 00 00 00 E0
DEC8 0E 00 01 02 04 08 04 00
DED0 00 00 80 40 20 20 30 00
DED8 00 00 00 00 18 3C 3C 18
DEE0 00 10 10 1E E0 00 0C 24
DEE8 42 82 43 00 00 00 00 00
DEF0 00 00 00 00 01 03 03 01
DEF8 00 00 00 00 80 C0 C0 80
DF00 00 01 01 01 0E 00 01 02
DF08 00 00 00 E0 00 00 80 40
DF10 04 08 04 00 00 00 00 00
DF18 20 20 30 00 00 00 00 00
DF20 1C 3E 7F FF FF FE FC 38
DF28 03 07 0F 0F 0F 07 03 01
DF30 80 C0 E0 F0 F0 F0 E0 C0
DF38 00 00 07 18 20 40 40 80
DF40 00 00 1F A0 C0 00 00 00
DF48 00 00 80 40 5C 22 02 02
DF50 80 40 7C 02 02 01 00 00
DF58 00 00 00 04 0A 11 60 00
DF60 02 04 08 04 04 04 F8 00
DF68 00 00 78 86 01 01 00 00
DF70 00 00 1E 61 80 80 00 00
DF78 00 00 00 00 87 79 00 00
DF80 00 00 00 00 E1 9E 00 00
DF88 22 14 08 08 08 08 08 08
DF90 18 3C 36 3E 7E 3C 18 18
DF98 18 18 0C 0C 06 06 03 03
DFA0 06 06 0C 0C 18 18 30 30
DFA8 60 60 C6 C3 66 6C 38 38
DFB0 84 D6 FF FF FF FF FF FF
DFB8 00 01 03 07 1F 3F 7F FF
DFC0 00 00 FF FF 3C 3C FF FF
DFC8 06 08 76 FF FF FF 3E 3C
DFD0 10 10 10 38 38 38 38 38
DFD8 10 1C 7C 38 38 38 10 10
DFE0 00 00 00 20 51 8A 04 00
DFE8 00 00 00 82 45 28 10 00
DFF0 00 00 00 00 00 00 00 00
DFF8 00 00 00 00 00 00 00 00

E000 00 00 00 00 00 00 00 00
E008 00 00 00 00 00 00 00 62
E010 00 B4 05 E9 00 C7 04 83
E018 00 3E 04 DC 00 C4 03 4E
E020 00 8C 03 93 00 C4 03 05
E028 01 3E 04 6E 00 11 05 83
E030 00 6A 06 31 00 B4 05 6E
E038 00 11 05 E9 00 C7 04 62
E040 00 B4 05 93 00 B4 05 2C
E048 00 6A 06 62 00 B4 05 DC
E050 00 11 05 5C 00 0C 06 DC
E058 00 A7 07 43 4C 49 46 46
E060 48 41 4E 47 45 52 43 52
E068 45 41 54 45 44 20 42 59
E070 20 41 2E 44 4F 45 57 52
E078 49 54 54 45 4E 20 42 59
E080 20 50 2E 43 4C 41 52 4B
E088 20 41 66 74 65 72 20 61
E090 20 73 68 6F 72 74 20 77
E098 61 6C 6B 20 57 69 6C 6C
E0A0 69 65 20 20 20 20 20 20
E0A8 72 65 74 75 72 6E 73 20
E0B0 74 6F 20 66 69 6E 64 20
E0B8 74 68 65 20 67 6F 61 74
E0C0 73 20 68 61 76 65 20 20
E0C8 73 70 72 65 61 64 29 68
E0D0 69 73 20 70 69 63 6E 69
E0D8 63 20 67 6F 6F 64 69 65
E0E0 73 20 61 6C 6C 20 20 20
E0E8 6F 76 65 72 20 61 20 72
E0F0 6F 63 6B 79 20 65 6D 62
E0F8 61 6E 6B 6D 65 6E 74 20
E100 20 20 20 20 20 20 20 20
E108 57 69 6C 6C 69 65 20 63
E110 65 74 73 20 6F 66 66 20
E118 74 6F 20 72 65 63 6C 61
E120 69 6D 20 68 69 73 20 20
E128 6C 6F 73 74 20 70 6F 73
E130 73 65 73 73 69 6F 6E 73
E138 2C 62 75 74 20 69 73 20
E140 68 61 6D 70 65 72 65 64
E148 62 79 20 66 61 6C 6C 69
E150 6E 67 20 62 6F 75 6C 64
E158 65 72 73 2C 70 6F 74 20
E160 68 6F 6C 65 73 20 20 20
E168 61 6E 64 20 76 69 63 69
E170 6F 75 73 20 73 6E 61 6B
E178 65 73 2E 54 6F 20 6D 61
E180 6B 65 20 20 20 20 20 20
```

- COMO DESCOBRIR OS ERROS
- TESTE O CONJUNTO COMPLETO
- VERIFICAÇÃO DA VERSÃO FINAL
- O CÓDIGO HEXADECIMAL
- ALTERAÇÕES POSSÍVEIS
- DESLOCAMENTO DO JOGO NA MEMÓRIA

```
E188 6D 61 74 74 65 72 73 20
E190 77 6F 72 73 65 20 74 68
E198 65 20 74 69 64 65 20 69
E1A0 73 20 72 69 73 69 6E 67
E1A8 61 6E 64 20 68 65 20 69
E1B0 73 20 69 6E 20 64 61 6E
E1B8 67 65 72 20 6F 66 20 62
E1C0 65 69 6E 67 20 63 75 74
E1C8 6F 66 66 2E 54 6F 20 68
E1D0 65 6C 70 20 57 69 6C 6C
E1D8 69 65 20 69 6E 20 68 69
E1E0 73 20 71 75 65 73 74 20
E1E8 72 65 61 64 20 74 68 65
E1F0 20 66 6F 6C 6C 6F 77 69
E1F8 6E 67 20 61 6E 64 20 70
E200 72 65 73 73 20 27 53 27
E208 74 6F 20 73 74 61 72 74
E210 2E 20 20 20 20 20 20 20
E218 20 20 20 20 20 20 20 20
E220 20 4E 20 20 20 20 2D 20
E228 52 75 6E 20 20 20 20 20
E230 20 20 20 20 20 20 20 20
E238 20 20 20 20 20 20 20 20
E240 20 4D 20 20 20 20 2D 20
E248 56 65 72 74 69 63 61 6C
E250 20 6A 75 6D 70 20 20 20
E258 20 20 20 20 20 20 20 20
E260 20 42 6F 74 68 20 2D 20
E268 44 69 61 67 6F 6E 61 6C
E270 20 6A 75 6D 70 53 43 4F
E278 52 45 2D 30 30 30 30 30
E280 30 4C 49 56 45 53 2D 35
E288 47 41 4D 45 20 4F 56 45
E290 52 20 21 21 21 23 23 21
E298 23 23 23 21 23 23 21 23
E2A0 21 23 23 23 23 21 23 21
```

```
E2A8 23 23 23 23 21 23 23 21
E2B0 23 23 23 CD 50 E3 3E 02
E2B8 D3 FE 3E 10 32 48 5C DD
E2C0 21 5B E0 06 05 3E 46 21
E2C8 86 00 CD 2B E3 06 06 21
E2D0 CC 00 CD 2B E3 06 10 3E
E2D8 07 21 62 02 CD 2B E3 06
E2E0 12 21 A2 02 CD 2B E3 06
E2E8 02 21 E8 FD 11 00 00 2B
E2F0 E5 ED 52 E1 20 F9 10 F1
E2F8 CD 50 E3 DD 21 88 E0 21
E300 20 00 3E 07 06 FF CD 2B
E308 E3 06 8A CD 2B E3 11 26
E310 00 19 06 64 3E 46 CD 2B
E318 E3 3E FD DB FE CB 4F 20
E320 F8 C9 3E FD DB FE CB 4F
E328 20 F8 C9 C5 F5 DD 7E 00
E330 CD 3E E3 F1 CD 69 E3 23
E338 DD 23 C1 10 EE C9 E5 21
E340 F8 3C 11 08 00 06 1F 90
E348 19 3D 20 FC E5 C1 E1 C9
E350 DD 21 00 40 21 00 1B 3E
E358 00 DD 77 00 DD 23 2B E5
E360 11 00 00 ED 52 E1 20 F1
E368 C9 F5 E5 C5 E5 D1 7A FE
E370 01 38 11 D5 11 00 07 19
E378 D1 7A FE 01 28 06 D5 11
E380 00 97 19 D1 D5 11 00 40
E388 19 D1 3E 08 C1 F5 0A 77
E390 24 03 F1 3D 28 03 F5 18
E398 F5 E1 F1 D5 11 00 58 19
E3A0 D1 77 D5 E1 C9 00 00 00
E3A8 00 00 00 00 00 00 00 00
E3B0 00 00 00 00 00 00 00 00
E3B8 00 00 00 00 00 00 00 3E
E3C0 10 32 F0 DF DD 21 B2 E2
E3C8 06 20 C5 DC 35 E4 3E 00
E3D0 32 F1 DF DD 7E 00 DD 2B
E3D8 FE 21 20 0D 05 3A F0 DF
E3E0 3D 32 F0 DF 3E 01 32 F1
E3E8 DF 3A F0 DF 47 21 1F 00
E3F0 3E 2D CD 48 E4 01 B0 DF
E3F8 3A F1 DF FE 01 20 03 01
E400 B8 DF 3E 2C CD 69 E3 3A
E408 F0 DF 47 3E 17 90 47 3E
E410 20 11 20 00 19 CD 48 E4
E418 C1 10 AF 21 31 00 06 0C
E420 3E 29 DD 21 75 E2 CD 2B
E428 E3 21 71 00 06 07 CD 2B
E430 E3 CD 57 E4 C9 21 00 40
E438 06 D8 0E 1F 23 7E 2B 77
E440 23 0D 20 F8 23 10 F3 C9
E448 C5 01 00 3D CD 69 E3 11
E450 20 00 19 C1 10 F2 C9 3A
E458 00 E0 21 B8 DF 47 04 11
E460 08 00 19 10 FD E5 C1 21
E468 BF 00 3E 3A CD 69 E3 3A
E470 00 E0 FE 00 28 0C F5 CD
E478 83 E4 F1 FE 01 28 03 CD
E480 A9 E4 C9 21 C9 01 CD 96
E488 E4 21 91 01 CD 96 E4 21
E490 3A 01 CD 96 E4 C9 06 04
E498 C5 01 00 3D 3E 2D CD 69
E4A0 E3 11 20 00 19 C1 10 F0
E4A8 C9 21 C9 01 CD BC E4 21
E4B0 91 01 CD BC E4 21 3A 01
E4B8 CD BC E4 C9 3E 04 01 90
E4C0 DF F5 3E 2B CD 69 E3 11
E4C8 20 00 19 F1 3D 20 F2 C9
E4D0 CD B3 E2 3E 05 32 FF DF
E4D8 3E 00 32 00 E0 21 00 00
E4E0 22 F9 DF 22 FB DF 22 FD
E4E8 DF 3E 13 32 65 EA 3E 06
E4F0 32 09 E0 21 E0 02 22 0A
E4F8 E0 21 82 00 22 01 E0 3E
E500 03 32 03 E0 3E 00 32 04
E508 E0 3E 02 32 05 E0 21 C1
E510 01 22 F4 DF 21 00 00 22
E518 F6 DF 3E 00 32 F8 DF 21
E520 DF 00 22 0C E0 3E 00 06
E528 05 32 06 E0 80 32 07 E0
E530 80 32 08 E0 CD BF E3 CD
E538 3B E6 21 77 00 3A FF DF
E540 06 30 80 CD 3E E3 3E 29
E548 CD 69 E3 CD 60 EA CD 11
E550 E7 CD 71 E6 CD AF E9 CD
E558 02 E6 CD AB E5 CD 7F E5
E560 3A FB DF EF 01 CA BC E9
E568 FE 02 CA 04 E9 06 32 3E
E570 FF 3D 20 FD 10 FB 3E FE
E578 DB FE CB 47 20 D0 C9 3A
E580 05 E0 3C CB 9F 32 05 E0
E588 01 68 DF FE 04 38 03 01
E590 78 DF 3E 2E 21 2A 00 C5
E598 CD 69 E3 23 CD 69 E3 C1
E5A0 21 44 00 CD 69 E3 23 CD
E5A8 69 E3 C9 3A 03 E0 3D 32
E5B0 03 E0 FE 00 28 01 C9 3E
E5B8 06 32 03 E0 3E 2D 01 00
E5C0 40 2A 01 E0 16 03 1E 02
E5C8 CD 5A E6 3A 04 E0 FE 00
E5D0 28 03 2B 18 01 23 22 01
E5D8 E0 01 38 DF 3E 2F 16 03
E5E0 1E 02 CD 5A E6 11 81 00
E5E8 ED 52 20 06 3E 00 32 04
E5F0 E0 C9 11 90 00 2A 01 E0
E5F8 ED 52 20 05 3E 01 32 04
E600 E0 C9 01 E0 DF 3A 09 E0
E608 CB 57 28 03 01 E8 DF 2A
E610 0A E0 3E 0F 16 20 D5 C5
E618 CD 69 E3 23 C1 D1 15 20
E620 F5 3A 09 E0 3D 32 09 E0
E628 20 10 3E 0A 32 09 E0 2A
E630 0A E0 11 20 00 ED 52 22
E638 0A E0 C9 21 37 00 DD 21
E640 F9 D7 06 06 C5 DD 7E 00
E648 06 30 80 CD 3E E3 3E 29
E650 CD 69 E3 23 DD 23 C1 10
E658 EB C9 E5 D5 E5 D5 CD 69
E660 E3 23 D1 15 20 F7 E1 11
E668 20 00 19 D1 1D 20 EC E1
E670 C9 3A 00 E0 FE 00 28 05
E678 FE 03 28 01 C9 3A 0E E0
E680 FE 01 28 55 2A 0C E0 01
E688 20 DF 3E 2A CD 69 E3 23
E690 3E 2D 01 00 3D CD 69 E3
E698 2A 0C E0 11 E0 01 ED 52
E6A0 28 5D 2A 0C E0 11 20 58
E6A8 19 7E FE 0F 28 51 FE 2D
E6B0 20 1A 2A 0C E0 01 00 3D
E6B8 3E 2D CD 69 E3 11 20 00
E6C0 19 22 0C E0 01 20 DF 3E
E6C8 2A CD 69 E3 2A 0C E0 2B
E6D0 22 0C E0 3E 01 32 0E E0
E6D8 C9 2A 0C E0 11 00 58 19
E6E0 7E FE 28 20 05 3E 02 32
E6E8 F8 DF 2A 0C E0 3E 2A 01
E6F0 28 DF CD 69 E3 23 CD 69
E6F8 E3 3E 00 32 0E E0 69 2A
E700 0C E0 01 00 3D 3E 2D CD
E708 69 E3 21 DF 00 22 0C E0
E710 C9 3A F7 DF FE 00 C2 C8
E718 E7 3A F6 DF FE 01 28 5C
E720 2A F4 DF 2B 01 00 40 3E
E728 2D 11 02 02 CD 5A E6 01
E730 A8 DE 3E 28 23 11 02 01
E738 CD 5A E6 11 40 58 19 7E
E740 FE 2D CA C2 E7 FE 2B 28
E748 79 FE 0F 28 75 3E 00 DB
E750 FE CB 57 20 0E 06 01 CB
E758 5F 20 02 06 81 78 32 F7
E760 DF 18 09 CB 5F 20 05 3E
E768 01 32 F6 DF 2A F4 DF 11
E770 BF 00 ED 52 30 05 3E 01
E778 32 F8 DF C9 11 03 00 21
E780 0C 06 CD B5 03 2A F4 DF
E788 11 21 58 19 7E FE 2B 28
E790 31 FE 2C 28 27 FE 2A 28
E798 29 11 20 00 19 7E FE 0F
E7A0 28 20 FE 2D 28 1C FE 2B
E7A8 28 18 2A F4 DF 3E 28 01
E7B0 B8 DE 11 02 02 CD 5A E6
E7B8 23 22 F4 DF 3E 00 32 F6
E7C0 DF C9 3E 02 32 F8 DF C9
E7C8 11 06 00 21 F7 02 CD B5
E7D0 03 3A F7 DF FE 01 20 1B
E7D8 3C 32 F7 DF 2A F4 DF 11
E7E0 20 00 ED 52 22 F4 DF 01
E7E8 D8 DE 3E 28 11 03 01 CD
E7F0 56 E6 C9 FE 02 20 1F 3C
E7F8 32 F7 DF 2A F4 DF 01 A8
E800 DE 3E 28 11 02 01 CD 5A
E808 E6 11 40 00 19 3E 2D 01
E810 00 3D CD 69 E3 C9 FE 03
E818 20 1A 3C 32 F7 DF 2A F4
E820 DF 01 D8 DE 3E 28 11 03
E828 01 CD 5A E6 11 20 00 19
E830 22 F4 DF C9 FE 04 20 18
E838 3E 00 32 F7 DF 2A F4 DF
E840 11 20 00 ED 52 01 00 3D
E848 3E 2D CD 69 E3 C3 11 E7
E850 FE 81 20 33 3C 32 F7 DF
E858 2A F4 DF 11 21 58 19 7E
E860 FE 2B CA C2 E7 FE 2C 20
E868 07 2A F4 DF 2B 22 F4 DF
E870 2A F4 DF 11 20 00 ED 52
E878 22 F4 DF 01 F0 DE 11 03
E880 02 3E 28 CD 5A E6 C9 FE
E888 84 28 5B 3C 32 F7 DF 3A
E890 F6 DF FE 01 28 23 2A F4
E898 DF 01 00 40 3E 2D 11 03
E8A0 02 CD 5A E6 23 22 F4 DF
E8A8 01 A8 DE 3E 28 11 02 01
E8B0 CD 5A E6 3E 01 32 F6 DF
E8B8 C9 2A F4 DF 01 B8 DE 3E
E8C0 28 11 02 02 CD 5A E6 23
E8C8 22 F4 DE 11 40 58 19 7E
E8D0 FE 2C 20 0C 3E 00 32 F7
E8D8 DF 3E 04 06 05 CD FC E9
E8E0 3E 00 32 F6 DF C9 3E 00
E8E8 32 F7 DF 2A F4 DF 2B 01
E8F0 00 40 3E 2D 11 02 02 CD
E8F8 5A E6 11 21 00 19 22 F4
E900 DF C3 11 E7 11 C4 00 21
E908 3E 04 CD B5 03 11 83 00
E910 21 6E 06 CD B5 03 2A F4
E918 DF 01 11 3D 3E 2D CD 69
E920 E3 11 20 00 19 22 F4 DF
E928 01 A8 DE 3E 28 11 02 01
E930 CD 5A E6 11 1E 00 21 6E
E938 03 ED 5F 6F CD B5 03 2A
E940 F4 DF 11 C0 02 ED 52 38
E948 CD 3E 09 32 65 EA 3A FF
E950 DF 3D 32 FF DF C2 EE E4
E958 21 4A 01 3E 8E 06 0B DD
E960 21 88 E2 CD 2B E3 CD 3B
E968 E6 11 05 01 21 6E 06 CD
E970 B5 03 11 88 01 21 3E 04
E978 CD B5 03 11 05 01 21 6E
E980 06 CD B5 03 3E 32 32 6E
E988 E5 C3 D0 E4 11 0B 02 21
E990 26 03 CD B5 03 3A 00 E0
```

```
E998 3C CB 97 32 00 E0 3A 6E
E9A0 E5 3D 32 6E E5 3E 03 06
E9A8 05 CD FC E9 C3 E9 E4 3A
E9B0 00 E0 FE 02 30 01 C9 DD
E9B8 21 06 E0 21 A9 01 CD CE
E9C0 E9 21 71 01 CD CE E9 21
E9C8 1A 01 CD CE E9 C9 E5 ED
E9D0 5B 0A E0 ED 52 E1 38 01
E9D8 C9 DD 7E 00 3C CB A7 DD
E9E0 77 00 DD 23 FE 07 30 01
E9E8 C9 01 88 DF 16 2B FE 0F
E9F0 20 05 01 00 3D 16 2D 7A
E9F8 CD 69 E3 C9 DD 21 F9 DF
EA00 16 00 5F DD 19 DD E5 CD
EA08 12 EA DD EA 10 F7 CD 3B
EA10 E6 C9 DD E7 00 3C FE 0A
EA18 20 09 3E 00 DD 77 00 DD
EA20 2B 18 EF DD 77 00 C9 00
EA28 00 00 00 00 00 00 00 00
EA30 00 00 00 00 00 00 00 00
EA38 00 00 00 00 00 00 00 00
EA40 00 00 00 00 00 00 00 00
EA48 00 00 00 00 00 00 00 00
EA50 00 00 00 00 00 00 00 00
EA58 00 00 00 00 00 00 00 00
EA60 DD 21 0F E0 06 13 C5 DD
EA68 56 01 DD 5E 00 DD 66 03
EA70 DD 6E 02 DD E5 CD B5 03
EA78 DD E1 11 04 00 DD 19 C1
EA80 10 E4 C9 00 00 00 00 00
```

T

Você já deve ter observado que a rotina da música funciona independentemente de qualquer outra e foi montada distante do programa principal. Isso não interfere com o Assembler de *INPUT*, mas, se você usou outro montador, poderá descobrir que a rotina da música está na região de memória ocupada pelo próprio Assembler. Nesse caso, coloque uma nova origem na rotina da música. Tente usar 24000 em vez de 30000 — e não se esqueça de mudar as declarações **EQU** e as instruções **JSR 30000**, para que o processador vá para o novo endereço inicial quando a rotina da música for chamada.

Use o monitor de código de máquina do seu computador para ler (com **PEEK**) a memória e verificar se ela confere com a listagem hexa dada a seguir.

```
4268 03 0C 09 06 06 08 01 0E
4270 07 05 12 03 12 05 01 14
4278 05 04 20 02 19 20 01 2E
4280 04 0F 05 17 12 09 14 14
4288 05 0E 20 02 19 20 13 2E
4290 0B 05 0C 0C 01 17 01 19
4298 01 0E 04 20 07 2E 08 05
42A0 04 0C 05 19 20 01 06 14
42A8 05 12 20 01 20 13 08 0F
42B0 12 14 20 17 01 0C 0B 20
42B8 17 09 0C 0C 09 05 20 20
42C0 20 20 20 20 12 05 14 05
42C8 12 0E 13 20 14 0F 20 06
42D0 09 0E 04 20 14 08 05 20
42D8 07 0F 01 14 13 20 08 01
42E0 16 05 20 20 13 10 12 05
42E8 01 04 20 08 09 13 20 10
42F0 09 03 0E 09 03 20 07 0F
42F8 0F 04 09 05 13 20 01 0C
4300 0C 20 20 20 0F 16 05 12
4308 20 01 20 12 0F 03 0B 19
4310 20 05 0D 02 01 0E 0B 0D
4318 05 0E 14 2E 20 20 20 20
4320 20 20 20 20 20 17 09 0C
4328 0C 09 05 20 13 05 14 13
4330 20 0F 06 06 20 14 0F 20
4338 12 05 03 0C 01 09 0D 20
4340 08 09 13 20 0C 0F 13 14
4348 20 10 0F 13 13 05 13 13
4350 09 0F 0E 13 2C 02 15 14
4358 20 09 13 20 08 01 0D 10
4360 05 12 05 04 02 19 20 06
4368 01 0C 0C 09 0E 07 20 02
4370 0F 15 0C 04 05 12 13 2C
4378 10 0F 14 20 08 0F 0C 05
4380 13 2C 20 20 01 0E 04 20
4388 16 09 03 09 0F 15 13 20
4390 13 0E 01 0B 05 13 2E 14
4398 0F 20 0D 01 0B 05 20 20
43A0 20 20 20 20 0D 01 14 14
43A8 05 12 13 20 17 0F 12 13
43B0 05 20 14 08 05 20 14 09
43B8 04 05 20 09 13 20 12 09
43C0 13 09 0E 07 01 0E 04 20
43C8 08 05 20 09 13 20 09 0E
43D0 20 04 01 0E 07 05 12 20
43D8 0F 06 20 02 05 09 0E 07
43E0 20 03 15 14 0F 06 06 2E
43E8 14 0F 20 08 05 0C 10 20
43F0 17 09 0C 0C 09 05 20 09
43F8 0E 20 08 09 13 20 11 15
4400 05 13 14 20 12 05 01 04
4408 20 14 08 05 20 06 0F 0C
4410 0C 0F 17 09 0E 07 20 01
4418 0E 04 20 10 12 05 13 13
4420 20 27 13 27 14 0F 20 13
4428 14 01 12 14 2E 0E 20 20
4430 20 20 2D 20 12 15 0E 0D
4438 20 20 20 20 2D 20 16 05
4440 12 14 09 03 01 0C 20 0A
4448 15 0D 10 02 0F 14 08 20
4450 2D 20 04 09 01 07 0F 0E
4458 01 0C 20 0A 15 0D 10 23
4460 23 23 21 23 23 21 23 23
4468 23 21 23 23 21 23 23 23
4470 23 21 23 23 21 23 23 21
4478 23 23 23 21 23 23 21 21
4480 55 54 50 50 40 40 00 00
4488 7F 5F 57 D7 F5 FF D5 75
4490 DD 77 5D D5 7D 75 5D 77
4498 F5 FD 57 75 DD 77 5D D5
44A0 FD 5F 57 D7 5D FF D5 7F
44A8 75 77 FD F7 D5 5D 75 77
44B0 55 D5 D5 5D 5D D7 F5 7D
44B8 D5 5D 5D D7 55 57 FF 7F
44C0 57 57 FD FD 55 75 D5 55
44C8 55 75 D5 55 75 75 D5 D5
44D0 5D 5D D5 D5 5D 5D D7 55
44D8 57 5D D7 5D D5 D5 5D 5D
44E0 75 D7 DD 75 7D 5D 75 D5
44E8 5D 75 5D D7 57 75 5D 5D
44F0 F5 D5 57 75 5D D5 57 75
44F8 57 55 55 D7 F7 55 55 DD
4500 57 55 55 D5 77 55 55 DF
4508 D5 D5 57 55 5D D5 57 7F
4510 F5 75 5D 55 57 75 5D F5
4518 57 5D 75 5F 5D 57 D7 55
4520 75 75 57 55 55 77 75 D5
4528 55 D7 75 D5 55 D7 75 75
4530 57 57 5D 75 57 57 5D 55
4538 55 57 5D 55 7D D7 D7 D7
4540 7D 5D 7D 5D 5D FF 7D D7
4548 5D 75 FF FD 57 FD 57 FD
4550 5D 7D DD FF 5D FF D5 7D
4558 57 FD 7F D5 FD D7 7D FF
4560 57 5D 75 75 7D D7 7D D7
4568 7D 7F D7 7F 57 57 57 5F
4570 5F 57 5F 5F 5F 5F D5 F5
4578 F5 D5 F5 F5 F5 F5 5F 5F
4580 57 57 57 57 57 57 F5 F5
4588 D5 D5 D5 D5 D5 FD 55 55
4590 55 55 55 55 55 55 57 5F
4598 5F 57 55 57 57 57 D5 F5
45A0 F5 D5 55 55 55 FD 55 55
45A8 55 55 55 55 55 55 55 55
45B0 55 55 55 55 55 55 FD 55
45B8 57 5D 75 D5 75 55 55 55
45C0 D5 75 5D 5D 5F 55 55 55
45C8 55 55 55 55 55 55 55 55
45D0 55 55 57 5F 5F 57 55 55
45D8 55 55 D5 F5 F5 D5 55 57
45E0 57 57 FD 55 57 5D 55 55
45E8 55 FD 55 55 D5 75 75 D5
45F0 75 55 55 55 55 55 5D 5D
45F8 5F 55 55 55 55 55 55 55
4600 55 55 55 55 55 55 55 55
4608 55 55 57 5F 5F 57 55 55
4610 55 55 D5 F5 F5 D5 55 55
4618 55 55 55 55 55 55 55 55
4620 55 55 55 55 55 55 55 57
4628 57 57 FD 55 57 5D 55 55
4630 55 FD 55 55 D5 75 55 55
4638 55 55 55 55 55 55 55 55
4640 55 55 55 55 55 55 75 D5
4648 75 55 55 55 55 55 5D 5D
4650 5F 55 55 55 55 55 55 55
4658 55 55 55 55 55 55 57 5F
4660 7F FF FF FF 7F 5F F5 FD
4668 FF FF FF FD F5 D5 55 55
4670 55 55 55 55 55 55 5F 7F
4678 FF FF FF 7F 5F 57 D5 F5
4680 FD FF FF FF FD F5 55 55
4688 55 55 55 55 55 55 5D 57
4690 55 55 55 55 55 55 5D 75
4698 D5 D5 D5 D5 D5 D5 57 5F
46A0 7D 7F 7F 5F 57 57 D5 F5
46A8 FD FD FD F5 D5 D5 57 57
46B0 55 55 55 55 55 55 D5 D5
46B8 F5 F5 7D 7D 5F 5F 55 55
46C0 55 55 57 57 5F 5F 7D 7D
46C8 F5 F5 D5 D5 55 55 7D 7D
46D0 F5 F5 7D 7D 5F 5F 55 55
46D8 7D 5F 7D F5 D5 D5 55 55
46E0 AA AA 56 56 AA AA 55 55
46E8 AA AA 95 95 AA AA 55 55
46F0 7F FF FF FF 7F 5F 81 15
46F8 7D FF FF FF FD F5 57 57
4700 57 5F 5F 5F 5F 5F 55 55
4708 55 D5 D5 D5 D5 D5 57 5F
4710 7F 5F 5F 5F 57 57 55 D5
4718 F5 D5 D5 D5 55 55 AA AA
4720 AA A6 99 6A AA AA AA AA
4728 AA AA A9 66 9A AA AA AA
4730 AA 6A 9A A6 A9 AA AA AA
4738 AA A6 99 6A AA AA 00 05
4740 00 00 00 00 00 00 06 1D
4A38 BD 4A CC 8E 04 21 10 8E
4A40 42 68 C6 05 BD 4A E6 8E
4A48 04 67 C6 06 BD 4A E6 8E
4A50 05 61 C6 10 BD 4A E6 8E
4A58 05 A0 C6 15 BD 4A E6 8E
```

```
4A60 05 C7 C6 0C BD 4A E6 86
4A68 05 8E FF FF 30 1F 26 FC
4A70 4A 26 F6 BD 4A CC 8E 04
4A78 00 10 8E 42 A4 5F BD 4A
4A80 E6 C6 89 BD 4A E6 30 88
4A88 18 C6 0A BD 4A E6 30 88
4A90 16 C6 14 BD 4A E6 30 0C
4A98 C6 14 BD 4A E6 BD 80 06
4AA0 81 53 26 F9 39 BD 4A EE
4AA8 BD 4A D9 8E 15 FF 10 8E
4AB0 44 5F C6 20 34 04 BD 4A
4AB8 FD BD 4B 12 35 04 5A 26
4AC0 F3 10 8E 44 C4 8E 06 21
4AC8 BD 4B 43 39 8E 04 00 86
4AD0 80 A7 80 8C 06 00 25 F9
4ADB 39 8E 06 00 86 55 A7 80
4AE3 8C 1E 00 25 F9 39 A6 A0
4AE8 A7 80 5A 26 F9 39 86 E5
4AF0 B7 FF 22 B7 FF C3 B7 FF
4AF8 C5 B7 FF C7 39 34 30 8E
4B00 06 00 10 8E 06 01 A6 A0
4B08 A7 80 8C 1D FF 25 F7 35
4B10 30 39 34 10 A6 A0 80 21
4B18 26 1C 35 10 30 89 FF 00
4B20 34 10 34 20 10 8E 44 80
4B28 C6 08 A6 A0 A7 84 30 88
4B30 20 5A 26 F6 35 20 6F 84
4B38 30 88 20 8C 1E 00 25 F6
4B40 35 10 39 C6 1E 34 04 C6
4B48 04 A6 A0 A7 80 5A 26 F9
4B50 30 88 1C 35 04 5A 26 ED
4B58 39 B6 47 3E C6 10 3D C3
4B60 46 DE 1F 03 8E 0A DE BD
4B68 4B CA B6 47 3E 27 0E 34
4B70 02 BD 4B 7E 35 02 81 01
4B78 27 03 BD 4B 9A 39 8E 13
4B80 E7 CE 0B FF BD 4B B6 8E
4B88 11 EF CE 0B FF BD 4B B6
4B90 8E 0E F9 CE 0B FF BD 48
4B98 B6 39 8E 13 E7 CE 46 9E
4BA0 BD 4B B6 8E 11 EF CE 46
4BA8 9E BD 4B B6 8E 0E F9 CE
4BB0 46 9E BD 4B B6 39 C6 04
4BB8 34 54 BD 4B CA 35 54 30
4BC0 89 01 00 33 C8 10 5A 26
4BC8 EF 39 C6 02 34 14 C6 08
4BD0 37 02 A7 84 30 88 20 5A
4BD8 26 F6 35 14 30 01 5A 26
4BE0 EB 39 BD 4A 38 86 05 B7
4BE8 47 3F 7F 47 3E 8E 47 40
4BF0 C6 06 6F 80 5A 26 FB 86
4BF8 06 B7 47 46 8E 1D 00 BF
4C00 47 47 8E 13 E0 BF 47 49
4C08 7F 47 4B 7F 47 4C 8E 0B
4C10 FE BF 47 4D 7F 47 4F 86
4C18 05 B7 47 50 86 0A B7 47
4C20 51 BD 4A A5 8E 07 0F CE
4C28 44 88 C6 05 86 03 37 20
4C30 10 AF 81 4A 26 F8 30 88
4C38 1A 5A 26 F0 BD 4C 77 8E
4C40 08 0F CE 44 A6 C6 05 86
4C48 03 37 20 10 AF 81 4A 26
4C50 F8 30 88 1A 5A 26 F0 B6
4C58 47 3F C6 05 3D C3 45 3C
4C60 1F 03 8E 08 16 C6 05 37
4C68 02 A7 84 30 88 20 5A 26
4C70 F6 BD 75 30 7E 51 C4 34
4C78 36 8E 47 40 C6 06 10 8E
4C80 07 16 A6 84 34 14 C5 01
4C88 26 24 C6 05 3D C3 45 3C
4C90 1F 01 34 20 C6 05 A6 80
4C98 A7 A4 31 A8 20 5A 26 F6
4CA0 35 20 31 21 35 14 30 01
4CA8 5A 26 D7 35 36 39 C6 05
4CB0 3D C3 45 3C 1F 01 34 20
4CB8 C6 05 A6 80 34 02 84 0F
4CC0 48 48 48 48 8A 05 A7 21
4CC8 35 02 44 44 44 44 8A 50
4CD0 A7 A4 31 A8 20 5A 26 E2
4CD8 35 20 31 22 20 C6 CE 47
4CE0 1E B6 47 46 85 02 27 03
4CE8 CE 47 2E BE 47 47 86 10
4CF0 34 42 BD 4B CA 35 42 4A
4CF8 26 F6 7A 47 46 26 0F 86
4D00 0A B7 47 46 BE 47 47 30
4D08 89 FF 00 BF 47 47 39 7A
4D10 47 52 26 30 86 05 27 47
4D18 52 13 8E 06 21 86 1E 34
4D20 02 86 02 1C FE 34 01 5F
4D28 35 01 66 85 34 01 5C C1
4D30 8E 26 F5 68 84 35 01 66
4D38 84 4A 26 E7 30 88 20 35
4D40 02 4A 26 DB 39 B6 47 3E
4D48 27 05 81 03 27 01 39 BE
4D50 47 4D CE 06 00 34 10 BD
4D58 4B CA 35 10 30 1F 8C 14
4D60 E0 27 4B BF 47 4D A6 84
4D68 81 AA 27 42 81 55 27 09
4D70 81 5D 27 05 86 02 27 47
4D78 4C 30 89 01 21 A6 84 81
4D80 AA 27 2B 81 55 26 06 30
4D88 88 DF BF 47 4D B6 47 54
4D90 27 02 BE 47 4D CE 46 76
4D98 BD 4B CA 7F 47 54 39 BE
4DA0 47 4D CE 46 5E BD 4B CA
4DA8 86 01 B7 47 54 39 BE 47
4DB0 4D CE 06 00 BD 4B CA 8E
4DB8 0B FE BF 47 4D 39 FC 47
4DC0 49 C4 1F C1 1E 26 05 86
4DC8 01 B7 47 4C B6 47 55 10
4DD0 26 00 D9 BE 47 49 30 89
4DD8 02 20 AE 84 8C 55 55 10
4DE0 27 00 BB 8C AA AA 10 27
4DE8 00 B4 8C 5F F5 10 27 00
4DF0 AD 5F 7F 47 5B 86 BF B7
4DF8 FF 02 B6 FF 00 B7 47 56
4E00 86 DF B7 FF 02 B6 FF 00
4E08 B7 47 57 81 FD 26 1A C6
4E10 01 B6 47 56 81 FD 26 02
4E18 C6 81 F7 47 55 B6 47 58
4E20 26 15 BE 47 49 34 10 20
4E28 3C B6 47 56 81 FD 26 EA
4E30 86 01 B7 47 58 20 E3 BE
4E38 47 49 CE 06 00 BD 4B CA
4E40 30 89 00 FE BD 4B CA BE
4E48 47 49 30 01 34 10 30 89
4E50 01 61 A6 84 81 D5 D7 44
4E58 81 FF 27 40 81 50 27 44
4E60 B6 47 4B 27 16 AE E4 CE
4E68 45 CE BD 4B CA AE E4 30
4E70 89 01 00 BD 4B CA 7F 47
4E78 4B 20 19 AE E4 CE 45 96
4E80 BD 4B CA AE E4 30 89 01
4E88 00 CE 45 B6 BD 4B CA 86
4E90 01 B7 47 4B AE E4 BF 47
4E98 49 35 10 39 35 10 86 02
4EA0 B7 47 4C 39 35 10 30 1F
4EA8 34 10 20 B9 BD 51 6F B6
4EB0 47 55 81 01 26 2C 7C 47
4EB8 55 BE 47 49 34 10 30 89
4EC0 01 00 CE 06 00 BD 4B CA
4EC8 35 10 30 89 FF 00 BF 47
4ED0 49 CE 45 96 BD 4B CA 30
4ED8 89 00 FE CE 45 B6 BD 4B
4EE0 CA 39 81 02 26 1F 7C 47
4EE8 55 BE 47 49 30 89 FF 00
```

```
4EF0 CE 45 CE BD 4B CA 30 89
4EF8 00 FE BD 4B CA 30 89 00
4F00 FE BD 4B CA 39 81 03 26
4F08 2F 7C 47 55 BE 47 49 30
4F10 89 FF 00 CE 06 00 BD 4B
4F18 CA 30 89 00 FE CE 46 06
4F20 BD 4B CA 30 89 00 FE CE
4F28 46 26 BD 4B CA 30 89 00
4F30 FE CE 46 46 BD 4B CA 39
4F38 81 04 26 25 7F 47 55 BE
4F40 47 49 34 10 CE 06 00 BD
4F48 4B CA 35 10 30 89 01 00
4F50 BF 47 49 CE 45 6E BD 4B
4F58 CA 30 89 00 FE BD 4B CA
4F60 39 81 81 26 30 7C 47 55
4F68 BE 47 49 30 89 01 22 A6
4F70 84 81 57 10 27 FF 27 BD
4F78 50 3F BE 47 49 30 89 FF
4F80 01 BF 47 49 CE 45 96 BD
4F88 4B CA 30 89 00 FE CE 45
4F90 B6 BD 4B CA 39 81 82 26
4F98 40 7C 47 55 BD 50 3F BE
4FA0 47 49 30 B9 FF 01 BF 47
4FA8 49 CE 45 CE BD 4B CA 30
4FB0 89 00 FE BD 4B CA 30 89
4FB8 00 FE BD 4B CA BE 47 49
4FC0 30 89 03 60 A6 80 81 FF
4FC8 27 07 A6 84 81 FF 27 01
4FD0 39 86 04 C6 05 BD 51 0F
4FD8 39 81 83 26 2B 7C 47 55
4FE0 BD 50 3F 30 89 00 FE CE
4FE8 06 00 BD 4B CA BE 47 49
4FF0 30 89 01 01 BF 47 49 CE
4FF8 45 96 BD 4B CA 30 89 00
5000 FE CE 45 B6 BD 4B CA 39
5008 81 84 26 1C BD 50 3F BE
5010 47 49 34 10 30 89 02 00
```

```
5018 10 AE 84 35 10 10 8C 55
5020 55 26 05 30 01 BF 47 49
5028 7F 47 55 7F 47 4B BE 47
5030 49 CE 45 6E BD 4B CA 30
5038 89 00 FE BD 4B CA 39 BE
5040 47 49 CE 06 00 BD 4B CA
5048 30 89 00 FE BD 4B CA 39
5050 86 88 8E 00 8C BD 51 33
5058 86 83 8E 00 D5 BD 51 33
5060 BE 47 49 CE 06 00 BD 4B
5068 CA 30 89 00 FE BF 47 49
5070 CE 45 6E BD 4B CA 30 89
5078 00 FE BD 4B CA 86 1E 8E
5080 00 71 BD 51 33 DE 47 49
5088 8C 1B 00 25 D3 7A 47 3F
5090 10 26 FB 63 B6 05 8E FF
5098 FF 30 1F 26 FC 4A 26 F6
50A0 BD 4A CC B6 FF 22 84 0F
50A8 B7 FF 22 B7 FF C2 B7 FF
50B0 C4 B7 FF C6 10 8E 05 0B
50B8 8E 07 01 AF A1 8E 0D 05
50C0 AF A1 8E 20 0F AF A1 8E
50C8 16 05 AF A1 8E 12 21 AF
50D0 A1 86 C8 8E 00 FF BD 51
50D8 33 86 C8 8E 00 C8 BD 51
50E0 33 86 FF 8E 00 FF BD 51
50E8 33 86 64 D7 51 EE 16 FA
50F0 F1 86 FF 8E 00 96 BD 51
50F8 33 B6 47 3E 4C 84 03 B7
5100 47 3E 7A 51 EE C6 05 86
5108 03 BD 51 0F 16 FA E8 1E
5110 89 8E 47 40 3A 34 12 BD
5118 51 23 35 12 4A 26 F2 BD
5120 4C 77 39 A6 84 4C 81 0A
5128 26 06 6F 84 30 1F 20 F3
5130 AF 84 39 34 02 B6 FF 01
5138 84 FF B7 FF 01 B6 FF 03
```

```
5140 84 F7 B7 FF 03 B6 FF 23
5148 8A 08 B7 FF 23 1A 50 35
5150 02 34 10 C6 FC F7 FF 20
5158 30 1F 26 FC AE E4 7F FF
5160 20 30 1F 26 FC AE E4 4A
5168 26 EB 1C AF 35 10 39 8E
5170 00 62 86 04 BD 51 33 39
5178 B6 47 E3 81 02 24 01 39
5180 10 8E 47 4F 8E 13 E7 BD
5188 51 97 8E 11 EF BD 51 97
5190 8E 0E F9 BD 51 97 39 34
5198 10 BE 47 47 30 88 1F AC
51A0 E4 35 10 22 01 39 A6 A4
51A8 4C 84 0F A7 A0 81 07 24
51B0 01 39 CE 46 8E 81 0F 26
51B8 03 CE 06 00 30 89 FF 00
51C0 BD 4B CA 39 86 05 B7 47
51C8 52 7F 47 55 BD 4B 59 BD
51D0 4D BE BD 4D 45 BD 51 78
51D8 BD 4C DE BD 4D 0F B6 47
51E0 4C 81 01 10 27 FF 0A 81
51E8 02 10 27 FE 63 C6 64 4F
51F0 4A 26 CD 5A 26 F9 BD 80
51F8 06 81 03 26 D2 39 39 03
5200 00 00 00 00 00 00 00 00
7530 8E 75 8A BF 75 88 86 13
7538 34 02 B6 FF 01 84 F7 B7
7540 FF 01 B6 FF 03 84 F7 B7
7548 FF 03 B6 FF 23 8A 08 B7
7550 FF 23 FE 75 88 1A 50 37
7558 12 11 83 75 CD 25 03 CE
7560 75 8A FF 75 88 34 10 C6
7568 FC F7 FF 20 30 1F 26 FC
7570 AE E4 7F FF 20 30 1F 26
7578 FC AE E4 4A 26 EB 32 62
7580 6A E4 26 CE 1C AF 35 82
7588 75 C3 62 00 BD E9 00 9E
```

```
7590 83 00 8D DC 00 7D 4E 00
7598 76 93 00 7D FF 00 8D 6E
75A0 00 A8 83 00 D4 31 00 BD
75A8 6E 00 A8 E9 00 9E 62 00
75B0 BD 93 00 BD 2C 00 D4 62
75B8 00 BD DC 00 A8 5C 00 C8
75C0 DC 00 FC 00 00 00 00 00
75C8 00 00 00 00 00 00 00 00
```

Avalanche foi montado a partir do endereço 49436, para possibilitar a utilização do Assembler de *INPUT*.

Provavelmente você já percebeu que alguns trechos da memória do computador não foram aproveitados: entre a tabela de cores e a tabela de padrões e entre o fim do programa e a tabela de perfil. Assim, você terá a possibilidade de alterar os endereços de montagem tanto da tabela de cores como da tabela de perfil, mas não se esqueça de alterar também todas as instruções nas rotinas que fazem referência a essas tabelas. Essa precaução deve ser tomada sempre que quiser modificar determinado endereço de montagem no programa.

Faça um programa BASIC ou examine manualmente as posições da memória, verificando se os conteúdos conferem com a listagem hexadecimal a seguir. Se você tiver deslocado alguma parte do jogo, os endereços serão diferentes, mas a seqüência de códigos hexa deve ser exatamente a mesma.

```
C11C 17 17 17 17 17 17 17 17
C124 17 17 17 17 17 17 17 17
C12C 17 17 17 17 17 17 17 17
C134 17 17 17 17 17 17 17 17
C13C 17 17 17 17 17 17 17 17
C144 17 17 17 17 17 17 17 17
C14C 17 17 17 17 17 17 17 17
C154 17 17 17 17 17 17 17 17
C15C 17 17 17 17 17 17 17 17
C164 17 17 17 17 17 17 17 17
C16C 17 17 17 17 17 17 17 17
C174 17 17 17 17 17 17 17 17
C17C 67 67 67 67 67 67 67 67
C184 67 67 67 67 67 67 67 67
C18C 67 67 67 67 67 67 67 67
C194 67 67 67 67 67 67 67 67
C19C 67 67 67 67 67 67 67 67
C1A4 67 67 67 67 67 67 67 67
C1AC 67 67 67 67 67 67 67 67
C1B4 67 67 67 67 67 67 67 67
C1BC F7 F7 F7 F7 F7 F7 F7 F7
C1C4 F7 F7 F7 F7 F7 F7 F7 F7
C1CC F7 F7 F7 F7 F7 F7 F7 F7
C1D4 F7 F7 F7 F7 F7 F7 F7 F7
C1DC F7 F7 F7 F7 F7 F7 F7 F7
C1E4 F7 F7 F7 F7 F7 F7 F7 F7
C1EC F7 F7 F7 F7 F7 F7 F7 F7
C1F4 F7 F7 F7 F7 F7 F7 F7 F7
C1FC 17 17 17 17 17 17 17 17
C204 17 17 17 17 17 17 17 17
C20C 17 17 17 17 17 17 17 17
C214 17 17 17 17 17 17 17 17
```

```
C21C 17 17 17 17 17 17 17 17
C224 17 17 17 17 17 17 17 17
C22C 17 17 17 17 17 17 17 17
C234 17 17 17 17 17 17 17 17
C23C C7 C7 C7 C7 C7 C7 C7 C7
C244 C7 C7 C7 C7 C7 C7 C7 C7
C24C C7 C7 C7 C7 C7 C7 C7 C7
C254 C7 C7 C7 C7 C7 C7 C7 C7
C25C C7 C7 C7 C7 C7 C7 C7 C7
C264 C7 C7 C7 C7 C7 C7 C7 C7
C26C C7 C7 C7 C7 C7 C7 C7 C7
C274 C7 C7 C7 C7 C7 C7 C7 C7
C27C C7 C7 C7 C7 C7 C7 C7 C7
C284 C7 C7 C7 C7 C7 C7 C7 C7
C28C C7 C7 C7 C7 C7 C7 C7 C7
C294 C7 C7 C7 C7 C7 C7 C7 C7
C29C 37 37 37 37 37 37 37 37
C2A4 37 37 37 37 37 37 37 37
C2AC 37 37 37 37 37 37 37 37
C2B4 37 37 37 37 37 37 37 37
C2BC 37 37 37 37 37 37 37 37
C2C4 37 37 37 37 37 37 37 37
C2CC 37 37 37 37 37 37 37 37
C2D4 37 37 37 37 37 37 37 37
C2DC A7 A7 A7 A7 A7 A7 A7 A7
C2E4 A7 A7 A7 A7 A7 A7 A7 A7
C2EC A7 A7 A7 A7 A7 A7 A7 A7
C2F4 A7 A7 A7 A7 A7 A7 A7 A7
C2FC 87 87 87 87 87 87 87 87
C304 87 87 87 87 87 87 87 87
C30C 87 87 87 87 87 87 87 87
C314 87 87 87 87 87 87 87 87
C31C D7 D7 D7 D7 D7 D7 D7 D7
C324 D7 D7 D7 D7 D7 D7 D7 D7
C32C D7 D7 D7 D7 D7 D7 D7 D7
C334 D7 D7 D7 D7 D7 D7 D7 D7
C33C 97 97 97 97 97 97 97 97
C344 97 97 97 97 97 97 97 97
C34C 97 97 97 97 97 97 97 97
C354 97 97 97 97 97 97 97 97
C35C F4 F4 F4 F4 F4 F4 F4 F4
C364 F4 F4 F4 F4 F4 F4 F4 F4
C36C F4 F4 F4 F4 F4 F4 F4 F4
C374 F4 F4 F4 F4 F4 F4 F4 F4
C37C F4 F4 F4 F4 F4 F4 F4 F4
C384 F4 F4 F4 F4 F4 F4 F4 F4
C38C F4 F4 F4 F4 F4 F4 F4 F4
C394 F4 F4 F4 F4 F4 F4 F4 F4
C39C 33 33 33 33 33 33 33 33
C3A4 33 33 33 33 33 33 33 33
C3AC 33 33 33 33 33 33 33 33
C3B4 33 33 33 33 33 33 33 33
C3BC F1 08 43 4C 41 20 00 00
C3C4 00 00 00 00 08 58 33 43
C3CC 24 30 00 00 00 00 00 08
C3D4 59 33 43 15 70 00 00 00
C3DC 00 00 03 42 00 01 28 F1
C3E4 00 00 FF FF 00 00 FF FF
C3EC 00 00 FF FF 00 00 FF FF
C3F4 00 00 FF FF 00 00 FF FF
C3FC 00 00 FF FF FF FF 00 00
C404 FF FF 00 00 FF FF 00 00
C40C FF FF 00 00 FF FF 00 00
C414 FF FF 00 00 FF FF 00 00
C41C FF FF 00 00 FF FF 00 00
C424 FF FF 00 00 FF FF 24 12
C42C 24 12 AE 0C 24 12 24 12
C434 AE 0C 02 0D 48 6C D3 00
C43C 54 5C 0F 24 12 AE 0C E8
C444 0C 48 6C D3 00 54 F0 F3
C44C 00 0C 00 48 6C 04 20 11
C454 00 2E FC 71 03 4B 07 18
C45C 01 CF 46 01 00 C8 7D F2
C464 2E BB 46 49 C5 2D 94 F3
C46C 6B 42 00 24 6B 2D 94 DB
C474 62 F3 62 01 46 00 00 00
C47C FF FF 00 00 FF FF 00 00
C484 FF FF 00 00 FF FF 00 00
C48C FF FF 00 00 FF FF 00 00
C494 FF FF 00 00 FF FF 00 00
C49C FF FF 00 00 18 3C 3C 18
C4A4 3C 3C 3C 3C 3C 3C 18 18
C4AC 18 18 18 1E 00 00 00 00
C4B4 00 00 00 00 00 00 00 00
C4BC 00 00 00 00 01 03 03 01
C4C4 00 01 01 01 0E 00 01 02
C4CC 04 08 04 00 80 C0 C0 80
C4D4 00 00 00 E0 00 00 80 40
C4DC 20 20 30 00 00 00 00 00
C4E4 00 00 00 00 00 00 00 00
C4EC 00 00 00 00 18 3C 3C 18
C4F4 00 10 10 1E E0 00 0C 24
C4FC 42 82 43 00 00 00 00 00
C504 00 00 00 00 1C 3E 7F FF
C50C FF FE FC 38 00 00 00 00
C514 00 00 00 00 00 00 00 00
C51C 00 00 00 00 00 00 00 00
C524 00 00 00 00 03 07 0F 0F
C52C 0F 07 03 01 00 00 00 00
C534 00 00 00 00 80 C0 E0 F0
C53C F0 F0 E0 C0 00 00 07 18
C544 20 40 40 80 80 40 7C 02
C54C 02 01 00 00 00 00 1F A0
C554 C0 00 00 00 00 00 00 04
C55C 0A 11 E0 00 00 00 80 40
C564 5C 22 02 02 02 04 08 04
C56C 04 04 F8 00 00 00 00 00
C574 00 00 00 00 00 00 00 00
C57C 00 00 00 00 00 00 78 86
C584 01 01 00 00 00 00 00 00
C58C 00 00 00 00 00 00 1E 61
C594 80 80 00 00 00 00 00 00
C59C 00 00 00 00 00 00 00 00
C5A4 87 79 00 00 00 00 00 00
C5AC 00 00 00 00 00 00 00 00
C5B4 E1 9E 00 00 00 00 00 00
C5BC 00 00 00 00 00 22 14 08 08
C5C4 08 08 08 08 18 3C 36 7E
C5CC 7E 3C 18 18 00 00 00 00
C5D4 00 00 00 00 00 00 00 00
C5DC 00 00 00 00 18 18 0C 0C
C5E4 06 06 03 03 06 06 0C 0C
C5EC 18 18 30 30 00 00 00 00
C5F4 00 00 00 00 00 00 00 00
C5FC 00 00 00 00 60 60 C6 C3
C604 66 6C 38 18 00 00 00 00
C60C 00 00 00 00 00 00 00 00
C614 00 00 00 00 00 00 00 00
C61C 00 00 00 00 84 D6 FF FF
C624 FF FF FF FF 00 00 00 00
C62C 00 00 00 00 00 00 00 00
C634 00 00 00 00 00 00 00 00
C63C 00 00 00 00 00 01 03 07
C644 1F 3F 7F FF 00 00 00 00
C64C 00 00 00 00 00 00 00 00
C654 00 00 00 00 00 00 00 00
C65C 00 00 00 00 00 00 FF FF
C664 3C 3C FF FF 00 00 00 00
C66C 00 00 00 00 00 00 00 00
C674 00 00 00 00 00 00 00 00
C67C 00 00 00 00 06 08 76 FF
C684 FF FF 7E 3C 00 00 00 00
C68C 00 00 00 00 00 00 00 00
C694 00 00 00 00 00 00 00 00
C69C 00 00 00 00 10 10 10 38
C6A4 38 38 38 38 00 00 00 00
C6AC 00 00 00 00 00 00 00 00
C6B4 00 00 00 00 00 00 00 00
C6BC 00 00 00 00 10 38 7C 38
C6C4 38 38 10 10 00 00 00 00
C6CC 00 00 00 00 00 00 00 00
C6D4 00 00 00 00 00 00 00 00
C6DC 00 00 00 00 00 00 00 20
C6E4 51 8A 04 00 00 00 00 00
C6EC 00 00 00 00 00 00 00 00
C6F4 00 00 00 00 00 00 00 00
C6FC 00 00 00 00 00 00 00 82
C704 45 28 10 00 00 00 00 00
C70C 00 00 00 00 00 00 00 00
C714 00 00 00 00 00 00 00 00
C71C 00 00 00 00 53 43 4F 52
C724 45 20 30 30 30 30 30 30
C72C 20 20 56 49 44 41 53 20
C734 30 20 20 4E 49 56 45 4C
C73C 20 30 4E 4F 56 41 20 50
C744 41 52 54 49 44 41 30 31
C74C 32 33 34 35 36 37 38 39
C754 00 00 00 A6 30 BA 30 95
C75C 07 18 08 0F 09 0F FD 00
C764 FD 00 0A D5 00 FD 00 1E
C76C BD 00 D5 00 0A A9 00 D5
C774 00 0F 9F 00 D5 00 05 A9
C77C 00 1C 01 0A BD 00 1C 01
C784 1E E1 00 E1 00 0A 1C 01
C78C E1 00 0F FD 00 E1 00 05
C794 E1 00 FD 00 0A D5 00 FD
C79C 00 1E FD 00 FD 00 0A FD
C7A4 00 3F 01 0F 0C 01 3F 01
C7AC 05 FD 00 3F 01 0A E1 00
C7B4 FD 00 11 E1 00 A9 00 0A
C7BC 0C 01 E1 00 0A A9 00 52
C7C4 01 14 FD 00 52 01 0A D5
C7CC 00 FD 00 1E BD 00 FD 00
C7D4 0A A9 00 D5 00 0F 9F 00
C7DC D5 00 05 A9 00 D5 00 0A
C7E4 BD 00 1C 01 1E E1 00 1C
C7EC 01 0A 1C 01 E1 00 0F FD
C7F4 00 E1 00 05 E1 00 E1 00
C7FC 0A D5 00 FD 00 1E E1 00
C804 E1 00 05 FD 00 D5 00 05
C80C FD 00 BD 00 05 0C 01 A9
C814 00 0F 2D 01 A9 00 05 0C
C81C 01 A9 00 0A FD 00 FD 00
C824 1E FD 00 7E 00 0A FD 00
C82C A9 00 0A FD 00 FD 00 1E
C834 8E 00 D5 00 1E 8E 00 BD
C83C 00 05 8E 00 A9 00 05 8E
C844 00 A9 00 0F 9F 00 BD 00
C84C 05 A9 00 D5 00 0A BD 00
C854 1C 01 1E BD 00 FD 00 05
C85C E1 00 E1 00 0A 1C 01 E1
C864 00 0F FD 00 FD 00 05 E1
C86C 00 1C 01 0A D5 00 FD 00
C874 1E D5 00 E1 00 05 FD 00
C87C D5 00 0A FD 00 D5 00 0F
C884 0C 01 E1 00 05 FD 00 FD
C88C 00 0A E1 00 52 01 11 E1
C894 00 A9 00 0A 0C 01 E1 00
C89C 0A 52 01 52 01 14 52 01
C8A4 7B 01 0A 8E 00 AA 01 1E
C8AC 8E 00 7B 01 05 8E 00 52
C8B4 01 0A 8E 00 52 01 0F 9F
C8BC 00 7B 01 05 A9 00 AA 01
C8C4 0A BD 00 1C 01 1E BD 00
C8CC FD 00 05 E1 00 E1 00 0A
C8D4 8E 00 E1 00 0F FD 00 FD
C8DC 00 05 E1 00 1C 01 0A D5
C8E4 00 FD 00 1E E1 00 E1 00
C8EC 05 FD 00 D5 00 05 FD 00
C8F4 BD 00 05 0C 01 A9 00 0F
C8FC 2D 01 A9 00 05 0C 01 52
C904 01 0A FD 00 FD 00 1E FD
```

```
C90C 00 7E 00 0A FD 00 A9 00
C914 0A FD 00 FD 00 1E 08 00
C91C 09 00 0A 00 64 68 6C 28
C924 2D 35 32 30 30 29 28 2D
C92C 35 32 30 30 29 2C 68 6C
C934 28 2D 35 32 30 30 29 2C
C93C 68 6C 6C 64 21 6C 64 6C
C944 64 35 35 35 35 35 32 32
C94C 35 35 32 35 41 56 41 4C
C954 41 4E 43 48 45 43 72 69
C95C 61 87 B1 6F 3A 52 2E 4E
C964 65 64 65 72 50 72 6F 67
C96C 72 61 6D 61 3A 4D 2E 48
C974 75 61 73 63 61 72 20 20
C97C 20 20 20 20 41 70 A2 73
C984 20 75 6D 20 70 65 71 75
C98C 65 6E 6F 20 70 61 73 73
C994 65 69 6F 2C 20 57 69 6C
C99C 69 65 20 20 20 20 20 20
C9A4 76 6F 6C 74 61 20 61 6F
C9AC 20 6C 6F 63 61 6C 20 6F
C9B4 6E 64 65 20 70 72 65 74
C9BC 65 6E 64 69 61 20 66 61
C9C4 7A 65 72 20 20 20 20 20
C9CC 73 65 75 20 70 69 71 75
C9D4 65 2D 6E 69 71 75 65 20
C9DC 65 20 64 65 73 63 6F 62
C9E4 72 65 20 71 75 65 20 75
C9EC 6D 20 20 20 20 20 20 20
C9F4 62 61 6E 64 6F 20 64 65
C9FC 20 63 61 62 72 69 74 6F
CA04 73 20 6D 6F 6E 74 65 73
CA0C 65 73 20 65 73 70 61 6C
CA14 68 6F 75 20 20 20 20 20
CA1C 74 6F 64 6F 20 73 65 75
CA24 20 6C 61 6E 63 68 65 20
CA2C 6E 61 20 65 6E 63 6F 73
CA34 74 61 2E 20 20 20 20 20
CA3C 20 20 20 20 20 20 20 20
CA44 20 20 20 20 20 20 20 20
CA4C 20 20 20 20 20 20 20 20
CA54 20 20 20 20 20 20 20 20
CA5C 20 20 20 20 20 20 20 20
CA64 20 20 20 20 20 20 20 20
CA6C 20 20 20 20 41 67 6F 72
CA74 61 20 65 6C 65 20 64 65
CA7C 76 65 20 73 75 62 69 72
CA84 20 61 6F 20 74 6F 70 6F
CA8C 20 64 61 20 20 20 20 20
CA94 6D 6F 6E 74 61 6E 68 61
CA9C 20 70 61 72 61 20 72 65
CAA4 63 75 70 65 72 61 72 20
CAAC 73 75 61 73 20 63 6F 69
CAB4 73 61 73 2C 20 20 20 20
CABC 65 6E 66 72 65 6E 74 61
CAC4 6E 64 6F 20 75 6D 61 20
CACC 61 76 61 6C 61 6E 63 68
CAD4 65 2C 20 63 6F 62 72 61
CADC 73 20 20 20 20 20 20 20
CAE4 76 65 6E 65 6E 6F 73 61
CAEC 73 20 65 20 63 6F 72 72
CAF4 65 6E 64 6F 20 6F 20 72
CAFC 69 73 63 6F 20 64 65 20
CB04 63 61 69 72 20 20 20 20
CB0C 6E 75 6D 20 62 75 72 61
CB14 63 6F 2E 20 20 20 20 20
CB1C 20 20 20 20 20 20 20 20
CB24 20 20 20 20 20 20 20 20
CB2C 20 20 20 20 20 20 20 20
CB34 20 20 20 20 20 20 20 20
CB3C 20 20 20 20 20 20 20 20
CB44 20 20 20 20 20 20 20 20
CB4C 20 20 20 20 20 20 20 20
CB54 20 20 20 20 20 20 20 20
CB5C 20 20 20 20 50 61 72 61
CB64 20 70 69 6F 72 61 72 20
CB6C 61 20 73 69 74 75 61 87
CB74 B1 6F 2C 20 61 20 6D 61
CB7C 72 82 20 20 20 20 20 20
CB84 65 73 74 A0 20 73 75 62
CB8C 69 6E 64 6F 20 65 20 65
CB94 6C 65 20 70 6F 64 65 20
CB9C 73 65 20 61 66 6F 67 61
CBA4 72 2E 20 20 20 20 20 20
CBAC 53 65 20 76 6F 63 88 20
CBB4 71 75 65 72 20 61 6A 75
CBBC 64 61 72 20 57 69 6C 6C
CBC4 69 65 2C 20 6C 65 69 61
CBCC 20 61 73 20 20 20 20 20
CBD4 69 6E 73 74 72 75 87 B5
CBDC 65 73 20 65 20 70 72 65
CBE4 73 73 69 6F 6E 65 20 61
CBEC 20 74 65 63 6C 61 20 27
CBF4 53 27 2E 20 20 20 20 20
CBFC 20 20 20 20 20 20 20 20
CC04 20 20 20 20 20 20 20 20
CC0C 20 20 20 20 20 20 20 20
CC14 20 20 20 20 20 20 20 20
CC1C 20 20 20 20 20 20 20 20
CC24 20 20 20 20 20 20 20 20
CC2C 20 20 20 20 20 20 20 20
CC34 20 20 20 20 20 20 20 20
CC3C 20 20 20 20 20 20 20 20
CC44 20 20 20 20 20 20 20 20
CC4C 55 73 65 3A 20 20 20 20
CC54 20 20 20 20 20 20 20 20
CC5C 20 20 20 20 20 20 20 20
CC64 20 20 20 20 20 20 20 20
CC6C 20 20 20 20 20 20 20 20
CC74 20 20 20 20 20 4E 20 20
CC7C 20 20 20 20 70 61 72 61
CC84 20 20 20 20 20 43 6F 72
CC8C 72 65 72 20 20 20 20 20
CC94 20 20 20 20 20 20 20 20
CC9C 20 20 20 20 20 4D 20 20
CCA4 20 20 20 20 70 61 72 61
CCAC 20 20 20 20 20 53 61 6C
CCB4 74 61 72 20 20 20 20 20
CCBC 20 20 20 20 20 20 20 20
CCC4 20 20 20 20 20 41 6D 62
CCCC 6F 73 20 20 70 61 72 61
CCD4 20 75 6D 20 20 53 61 6C
CCDC 74 6F 20 44 69 61 67 6F
CCE4 6E 61 6C 20 20 20 20 20
CCEC 20 20 20 20 20 20 20 20
CCF4 20 20 20 20 20 20 20 20
CCFC 20 20 20 20 20 20 20 20
CD04 20 20 20 20 20 20 20 20
CD0C 20 20 20 20 20 20 20 20
CD14 20 20 20 20 20 20 20 20
CD1C 20 20 20 20 20 20 20 20
CD24 20 20 20 20 20 20 20 20
CD2C 20 20 20 20 20 20 20 20
CD34 20 20 20 20 20 20 20 20

D000 CD 6C 00 11 28 01 21 50
D008 C9 01 09 00 CD 5C 00 11
D010 ED 01 21 59 C9 01 0F 00
D018 CD 5C 00 11 3C 02 21 68
D020 C9 01 12 00 CD 5C 00 06
D028 0A ED 6B 11 00 00 2B E5
D030 ED 52 E1 20 F9 10 F2 11
D038 00 00 21 7A C9 01 C0 03
D040 CD 5C 00 CD 9F 00 FE 53
D048 20 F9 C9 21 5C C7 06 06
D050 0E A0 ED A3 0E A1 ED A3
D058 20 F6 C9 06 04 0E A1 3E
D060 00 D3 A0 3C ED A3 20 F9
D068 C9 46 21 E8 03 11 00 00
D070 2B E5 ED 52 E1 20 F9 10
D078 F1 C9 CD 4B D0 06 14 C5
D080 CD 5B D0 E5 CD 69 D0 E1
D088 23 C1 10 F3 CD 4E D0 C9
D090 21 E9 F3 36 04 23 36 07
D098 23 36 07 CD 72 00 3E E2
D0A0 32 E0 F3 CD 69 00 C9 ED
D0A8 5B CB F3 21 A0 C4 01 80
D0B0 02 C5 E5 D5 CD 5C 00 D1
D0B8 21 00 08 19 54 5D E1 C1
D0C0 C5 E5 D5 CD 5C 00 D1 21
D0C8 00 08 19 54 5D E1 C1 C5
D0D0 E5 CD 5C 00 ED 5B CF F3
D0D8 E1 C1 CD 5C 00 C9 ED 5B
D0E0 C9 F3 21 1C C1 01 A0 02
D0E8 C5 E5 D5 CD 5C 00 D1 21
D0F0 00 08 19 54 5D E1 C1 C5
D0F8 E5 D5 CD 5C 00 D1 21 00
D100 08 19 54 5D E1 C1 CD 5C
D108 00 C9 2A C7 F3 11 90 E8
D110 01 00 03 CD 59 00 C9 11
D118 8F EB 21 8E EB 06 18 C5
D120 1A 01 1F 00 ED B8 12 2B
D128 1B C1 10 F3 C9 ED 5B C7
D130 F3 21 90 E8 01 00 03 CD
D138 5C 00 C9 3E FF 2A C7 F3
D140 01 00 03 CD 56 00 CD A7
D148 D0 CD DE D0 CD 0A D1 3E
D150 08 32 92 EB 3E 00 32 93
D158 EB 21 8F E8 06 21 C5 E5
D160 CD 2D D1 CD 17 D1 E1 3A
D168 93 EB FE 01 20 07 3A 92
D170 EB 3C 32 92 EB 3E 00 32
D178 93 EB 7E 2B E5 FE 21 20
D180 05 3E 01 32 93 EB 3A 92
D188 EB 47 21 90 E8 3E FF CD
D190 B5 D1 06 30 3A 93 EB FE
D198 01 20 02 06 34 78 77 3A
D1A0 92 EB 47 3E 17 90 47 3E
D1A8 51 11 20 00 19 CD B5 D1
D1B0 E1 C1 10 AA C9 77 11 20
D1B8 00 19 10 F9 C9 3A 94 EB
D1C0 17 17 06 38 80 2A C7 F3
D1C8 11 DE 00 19 CD 4D 00 3A
D1D0 94 EB FE 00 28 0C F5 CD
D1D8 E3 D1 F1 FE 01 28 03 CD
D1E0 0A D2 C9 11 C9 01 D5 11
D1E8 91 01 D5 11 3A 01 D5 0E
D1F0 03 2A C7 F3 D1 06 04 19
D1F8 3E FE E5 C5 CD 4D 00 C1
D200 E1 11 20 00 10 F1 0D 20
D208 E8 C9 3E 25 32 95 EB C6
D210 03 32 96 EB C6 01 32 97
D218 EB C6 03 32 98 EB 11 C9
D220 01 D5 11 91 01 D5 11 3A
D228 01 D5 0E 03 2A C7 F3 11
D230 95 EB ED 53 99 EB D1 06
D238 04 19 ED 5B 99 EB 1A 13
D240 ED 53 99 EB E5 C5 CD 4D
D248 00 C1 E1 11 20 00 10 E9
D250 0D 20 D9 C9 00 00 00 00
D258 00 00 CD 00 D0 3E 58 32
D260 7E D0 CD 7A D0 3E 05 32
D268 9B EB 3E 00 32 94 EB 21
D270 00 00 22 9D EB 22 9F EB
D278 22 A1 EB 3E 14 32 7E D0
D280 CD 90 D0 3E 06 32 A3 EB
D288 21 E0 02 22 A4 EB 21 82
D290 00 22 A6 EB 3E 03 32 A8
D298 EB 3E 00 32 A9 EB 3E 02
```

```
D2A0 32 AA EB 21 C2 01 22 AB
D2A8 EB 21 00 00 22 AD EB 3E
D2B0 00 32 AF EB 21 FF 00 22
D2B8 B0 EB 3E 00 06 05 32 B2
D2C0 EB 80 32 B3 EB 80 32 B4
D2C8 EB CD 3B D1 CD BD D1 3E
D2D0 A0 32 B7 FC 3E 10 32 B9
D2D8 FC 11 20 C7 06 34 C5 D5
D2E0 1A CD 8D 00 D1 13 C1 10
D2E8 F5 11 01 00 3E 54 06 1E
D2F0 C5 D5 F5 2A C7 F3 19 CD
D2F8 4D 00 F1 3C D1 13 C1 10
D300 EF CD 07 D3 C3 2E D3 11
D308 9D EB ED 53 BA EB 11 07
D310 00 2A C7 F3 19 06 06 C5
D318 E5 ED 5B BA EB 1A 13 ED
D320 53 BA EB C6 7E CD 4D 00
D328 E1 23 C1 10 EA C9 3A 9B
D330 EB C6 7E 11 15 00 2A C7
D338 F3 19 CD 4D 00 3A 94 EB
D340 C6 7E 11 1E 00 2A C7 F3
D348 19 CD 4D 00 CD 7A D0 21
D350 1A C9 CD 4E D0 CD 4B D5
D358 CD 80 D4 CD 0C D9 CD CA
D360 D3 CD FE D3 CD 8C D3 3A
D368 AF EB FE 01 CA D2 D8 FE
D370 02 CA 26 D8 06 32 3E FF
D378 3D 20 FD 10 F9 3E 07 CD
D380 41 01 CB 67 20 CF C9 C9
D388 C9 C9 FF FF 3A AA EB 3C
D390 CB 9F 32 AA EB 06 1C FE
D398 04 38 02 06 20 C5 78 2A
D3A0 C7 F3 11 2A 00 19 E5 F5
D3A8 CD 4D 00 F1 E1 23 C6 02
D3B0 CD 4D 00 C1 78 2A C7 F3
D3B8 11 44 00 19 E5 F5 CD 4D
D3C0 00 F1 E1 23 C6 02 CD 4D
D3C8 00 C9 06 48 3A A3 EB CB
D3D0 57 28 02 06 4C 2A C7 F3
D3D8 ED 5B A4 EB 19 78 01 20
D3E0 00 CD 56 00 3A A3 EB 3D
D3E8 32 A3 EB 20 10 3E 0A 32
D3F0 A3 EB 2A A4 EB 11 20 00
D3F8 ED 52 22 A4 EB C9 3A A8
D400 EB 3D 32 A8 EB FE 00 28
D408 01 C9 3E 06 32 A8 EB 3A
D410 A9 EB FE 00 28 06 21 A6
D418 EB 35 18 04 21 A6 EB 34
D420 2A CD F3 06 06 3A A6 EB
D428 4F 16 14 1E 01 CD 60 D4
D430 2A CD F3 11 04 00 19 06
D438 06 3A A6 EB C6 10 4F 16
D440 18 1E 01 CD 60 D4 3A A6
D448 EB FE 02 20 06 3E 00 32
D450 A9 EB C9 3A A6 EB FE E6
D458 20 05 3E 01 32 A9 EB C9
D460 78 D5 C5 E5 CD 4D 00 E1
D468 C1 23 79 E5 CD 4D 00 E1
D470 D1 23 7A D5 E5 CD 4D 00
D478 E1 D1 23 7B CD 4D 00 C9
D480 3A 94 EB FE 00 28 05 FE
D488 03 28 01 C9 3A B5 EB FE
D490 01 28 67 2A C7 F3 ED 5B
D498 B0 EB 19 3E 0D E5 CD 4D
D4A0 00 E1 23 3E FF CD 4D 00
D4A8 2A B0 EB 11 E0 01 ED 52
D4B0 28 4C 2A C7 F3 ED 5B B0
D4B8 EB 19 11 20 00 19 CD 4A
D4C0 00 FE 48 28 39 FE 4C 28
D4C8 35 FE FF 20 20 2A C7 F3
D4D0 ED 5B B0 EB 19 3E FF D5
D4D8 CD 4D 00 D1 21 20 00 19
D4E0 22 B0 EB ED 5B C7 F3 19
D4E8 3E 0D CD 4D 00 2A B0 EB
D4F0 2B 22 B0 EB 3E 01 32 B5
D4F8 EB C9 CD 02 D5 C9 CD 37
D500 D5 C9 2A C7 F3 ED 5B B0
D508 EB 19 E5 CD 4A 00 FE 00
D510 28 0C FE 01 28 08 FE 07
D518 28 04 FE 0B 20 05 3E 02
D520 32 AF EB E1 3E 11 E5 CD
D528 4D 00 E1 23 3E 13 CD 4D
D530 00 3E 00 32 B5 EB C9 2A
D538 C7 F3 ED 5B B0 EB 19 3E
D540 FF CD 4D 00 21 FF 00 22
D548 B0 EB C9 3A AE EB FE 00
D550 C2 46 D6 3A AD EB FE 01
D558 CA D3 D5 2A C7 F3 ED 5B
D560 AB EB 19 E5 2B 3E FF E5
D568 CD 4D 00 E1 11 20 00 19
D570 3E FF CD 4D 00 E1 3E 00
D578 E5 CD 4D 00 E1 11 20 00
D580 19 3E 01 E5 CD 4D 00 E1
D588 11 20 00 19 CD 4A 00 FE
D590 48 CA 40 D6 FE 4A CA 40
D598 D6 FE 25 CA 40 D6 FE FE
D5A0 CA 40 D6 3E 04 CD 41 01
D5A8 CB 57 20 0E 06 01 CB 5F
D5B0 20 02 06 81 78 32 AE EB
D5B8 18 09 CB 5F 20 05 3E 01
D5C0 32 AD EB 2A AB EB 11 DE
D5C8 00 ED 52 20 05 3E 01 32
D5D0 AF EB C9 2A C7 F3 ED 5B
D5D8 AB EB 19 11 21 00 19 E5
D5E0 CD 4A 00 E1 FE 24 28 58
D5E8 FE 34 28 4E FE 0D 28 50
D5F0 FE 11 28 4C 11 20 00 19
D5F8 CD 4A 00 FE 48 28 41 FE
D600 4A 28 3D FE FE 28 39 FE
D608 25 28 35 2A C7 F3 ED 5B
D610 AB EB 19 3E 04 E5 CD 4D
D618 00 E1 23 3E 06 E5 CD 4D
D620 00 E1 11 20 00 19 3E 07
D628 E5 CD 4D 00 E1 2B 3E 05
D630 CD 4D 00 2A AB EB 23 22
D638 AB EB 3E 00 32 AD EB C9
D640 3E 02 32 AF EB C9 3A AE
D648 EB FE 01 20 30 3C 32 AE
D650 EB 2A AB EB 11 20 00 ED
D658 52 22 AB EB ED 5B C7 F3
D660 19 3E 0A E5 CD 4D 00 E1
D668 11 20 00 19 3E 0B E5 CD
D670 4D 00 E1 11 20 00 19 3E
D678 FF CD 4D 00 C9 FE 02 20
D680 1D 3C 32 AE EB 2A AB EB
D688 ED 5B C7 F3 19 3E 00 E5
D690 CD 4D 00 E1 11 20 00 19
D698 3E 01 CD 4D 00 C9 FE 03
D6A0 20 1D 3C 32 AE EB 2A AB
D6A8 EB ED 5B C7 F3 19 3E 0A
D6B0 E5 CD 4D 00 E1 11 20 00
D6B8 19 3E 0B CD 4D 00 C9 FE
D6C0 04 20 1E 3E 00 32 AE EB
D6C8 2A AB EB E5 ED 5B C7 F3
D6D0 19 3E FF CD 4D 00 E1 11
D6D8 20 00 19 22 AB EB C3 4B
D6E0 D5 FE 81 20 6B 3C 32 AE
D6E8 EB 2A C7 F3 ED 5B AB EB
D6F0 19 11 21 00 19 CD 4A 00
D6F8 FE 24 CA 40 D6 FE 34 20
D700 07 2A AB EB 2B 22 AB EB
D708 2A AB EB ED 5B C7 F3 19
D710 E5 11 20 00 19 3E FF E5
D718 CD 4D 00 E1 23 3E FF CD
D720 4D 00 E1 3E 05 E5 CD 4D
D728 00 E1 23 3E 07 E5 CD 4D
D730 00 E1 11 20 00 ED 52 3E
D738 06 E5 CD 4D 00 E1 2B 3E
D740 04 CD 4D 00 2A AB EB 11
D748 20 00 ED 52 22 AB EB C9
D750 FE 84 CA EB D7 3C 32 AE
D758 EB 3A AD EB FE 01 CA 9A
D760 D7 2A AB EB ED 5B C7 F3
D768 19 3E FF E5 CD 4D 00 E1
D770 11 20 00 19 3E FF E5 CD
D778 4D 00 E1 23 3E 01 E5 CD
D780 4D 00 E1 11 20 00 ED 52
D788 3E 00 CD 4D 00 2A AB EB
D790 23 22 AB EB 3E 01 32 AD
D798 EB C9 2A AB EB ED 5B C7
D7A0 F3 19 3E 04 E5 CD 4D 00
D7A8 E1 23 3E 06 E5 CD 4D 00
D7B0 E1 11 20 00 19 3E 07 E5
D7B8 CD 4D 00 E1 2B 3E 05 CD
D7C0 4D 00 2A AB EB 23 22 AB
D7C8 EB ED 5B C7 F3 19 11 40
D7D0 00 19 CD 4A 00 FE 34 20
D7D8 0C 3E 00 32 AE EB 3E 04
D7E0 06 05 CD EC D8 3E 00 32
D7E8 AD EB C9 3E 00 32 AE EB
D7F0 2A AB EB ED 5B C7 F3 19
D7F8 E5 3E FF CD 4D 00 E1 2B
D800 3E FF E5 CD 4D 00 E1 11
D808 20 00 19 3E FF E5 CD 4D
D810 00 E1 23 3E FF CD 4D 00
D818 2A AB EB 11 20 00 19 22
D820 AB EB C3 4B D5 C9 2A AB
D828 EB ED 5B C7 F3 19 23 3E
D830 FF E5 CD 4D 00 E1 2B 2B
D838 3E FF E5 CD 4D 00 E1 11
D840 20 00 19 3E FF E5 CD 4D
D848 00 E1 23 23 3E FF E5 CD
D850 4D 00 E1 23 3E FF CD 4D
D858 00 2A AB EB E5 ED 5B C7
D860 F3 19 3E FF CD 4D 00 E1
D868 11 20 00 19 22 AB EB ED
D870 5B C7 F3 19 3E 00 E5 CD
D878 4D 00 E1 11 20 00 19 3E
D880 01 CD 4D 00 06 FF 3E FF
D888 3D 20 FD 10 F9 2A AB EB
D890 11 C0 02 ED 52 38 C2 3A
D898 9B EB 3D 32 9B EB C2 80
D8A0 D2 11 EA 00 3E 72 06 0C
D8A8 C5 D5 F5 2A C7 F3 19 CD
D8B0 4D 00 F1 3C D1 13 C1 10
D8B8 EF 06 1E 3E FF 0E FF 0D
D8C0 20 FD 3D 20 F8 10 F4 CD
D8C8 07 D3 3E 32 32 75 D3 C3
D8D0 5A D2 3A 94 EB 3C CB 97
D8D8 32 94 EB 3A 75 D3 3D 32
D8E0 75 D3 3E 03 06 05 CD EC
D8E8 D8 C3 80 D2 21 9D EB 16
D8F0 00 5F 19 E5 CD FE D8 E1
D8F8 10 F9 CD 07 D3 C9 7E 3C
D900 FE 0A 20 06 3E 00 77 2B
D908 18 F4 77 C9 3A 94 EB FE
D910 02 30 01 C9 01 B2 EB 21
D918 A9 01 CD 2A D9 21 71 01
D920 CD 2A D9 21 1A 01 CD 2A
D928 D9 C9 E5 ED 5B A4 EB ED
D930 52 E1 38 01 C9 0A 3C CB
D938 A7 02 03 FE 07 30 01 C9
D940 ED 5B C7 F3 19 FE 0F 06
D948 24 20 02 06 FF 78 CD 4D
D950 00 C9 00 00 00 00 00 00
E870 21 21 23 23 21 23 23 23
E878 21 23 23 21 23 21 23 23
E880 23 23 21 23 21 23 23 23
E888 23 21 23 23 21 23 23 23
```

AVALANCHE

ERRATA PARA O MSX

Para que o jogo *Avalanche* funcione perfeitamente no micro MSX, é preciso fazer algumas correções nas rotinas já publicadas. Apresentaremos essas correções por artigo.

EFEITOS SONOROS (pág. 788)

Neste artigo, tivemos quatro rotinas: a primeira programava o **PSG**, a segunda emitia os acordes, a terceira controlava a duração desses acordes e a quarta e principal chamava as anteriores na hora certa, tocando as 88 notas e desligando o **PSG**.

A primeira dessas rotinas foi montada apagando os três últimos bytes da rotina do artigo anterior, que coloca as instruções do jogo na tela. Com isso, o programa não funcionaria conforme estava previsto.

A fim de corrigir essa falha, a origem de montagem dessa primeira rotina passou de -12216 para -12213. Você pode pensar que terá de alterar a origem de todas as rotinas seguintes, mas, calma, isso não será preciso!

A segunda rotina inicia no endereço -12200, mas a sua primeira instrução, **ld hl,-14494**, é completamente desnecessária, uma vez que esse valor já está em HL quando essa rotina é chamada. Assim, basta suprimir a instrução, e a origem da segunda rotina passa a ser -12197. Feito isso, não precisamos alterar mais nenhuma origem.

Completando, devemos atualizar essas novas origens na quarta rotina, onde elas são chamadas. Na linha 20, a instrução **call -12216** passa a ser **call -12213** e, na linha 120, a instrução **call -12213** passa a ser **call -12210**.

MONTE O CENÁRIO (pág. 824)

Na linha 30 da segunda rotina deste artigo, que inicia na posição -12121, a instrução **ld hl,-15100** passa a ser **ld hl,-15200**.

RISCOS E PRÊMIOS (pág. 941)

Neste artigo, a origem deve ser alterada de 53682 para 53693.

Na linha 80, a instrução **ld de,254** passa a ser **ld de,222**; isto é feito para que a pedra não apague o item do lanche no topo da montanha.

Na linha 600, a instrução **ld(-5220),de** passa a ser **ld(-5223),de**. Na linha 640, **ld de,(-5220)** passa a ser **ld de,(-5223)** e, na linha 670, **ld (-5220),de** passa para **ld de,(-5223)**.

O endereço -5220 foi trocado pelo endereço -5223 porque as instruções afetam os endereços -5220 e -5219, e este último é utilizado para armazenar um dígito do escore.

No texto deste artigo, onde se lê **hls**, leia-se **ho**; onde se lê **snp**, leia-se **sn**; onde se lê **ho**, leia-se **po**; onde se lê **hp**, leia-se **pp**; onde se lê **pusch**, leia-se **push**. Com essas mudanças, o texto concorda com os rótulos usados no programa.

ROTINA PRINCIPAL (pág. 969)

Neste artigo, a origem da rotina precisa ser alterada: ela passa de -11670 para -11678. Guarde bem esse novo endereço, pois você irá usá-lo para dar início ao jogo. Abaixo da linha 20, vamos inserir duas linhas: 25 **ld a,88** e 28 **ld (-12162),a**. Isto é feito a fim de ajustar a rotina da música para tocar todas as 88 notas. Acrescente ainda a linha 135 **call -12144** abaixo da linha 130 **ld (12162),a**. Esta instrução é adicionada para executar a rotina de inicialização do VDP.

ACERTO DAS VARIÁVEIS (pág. 995)

Na linha 140, a instrução **ld hl,481** passa a ser **ld hl,450**. Esta instrução ajusta a posição inicial de Willie.

AS PEDRAS ROLAM (2) (pág. 1128)

Na linha 90, a instrução **cp 5** deve ser alterada para **cp 0**.

MAIS SALTOS (pág. 1186)

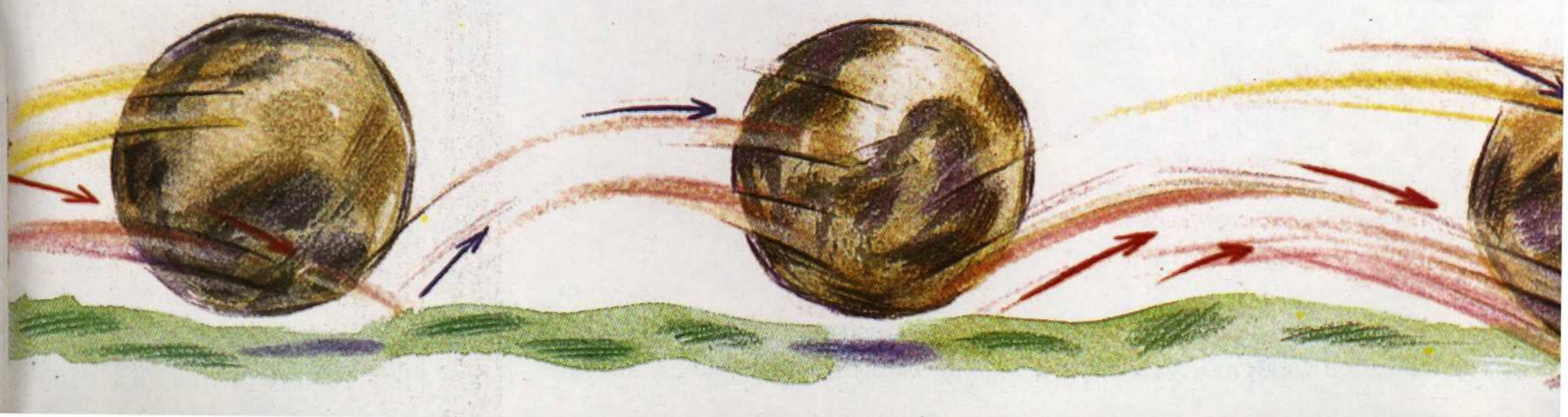
Na linha 350, do segundo programa deste artigo, altere a instrução **ld a,3** para **ld a,4**.

AS CINCO VIDAS DE WILLIE (pág. 1208)

Na linha 900, a instrução **jp 53855** deve ser corrigida para **jp 53850**.

PONTOS GANHOS (pág. 1228)

Na linha 90, a instrução **ld a,2** deve ser alterada para **ld a,3**.

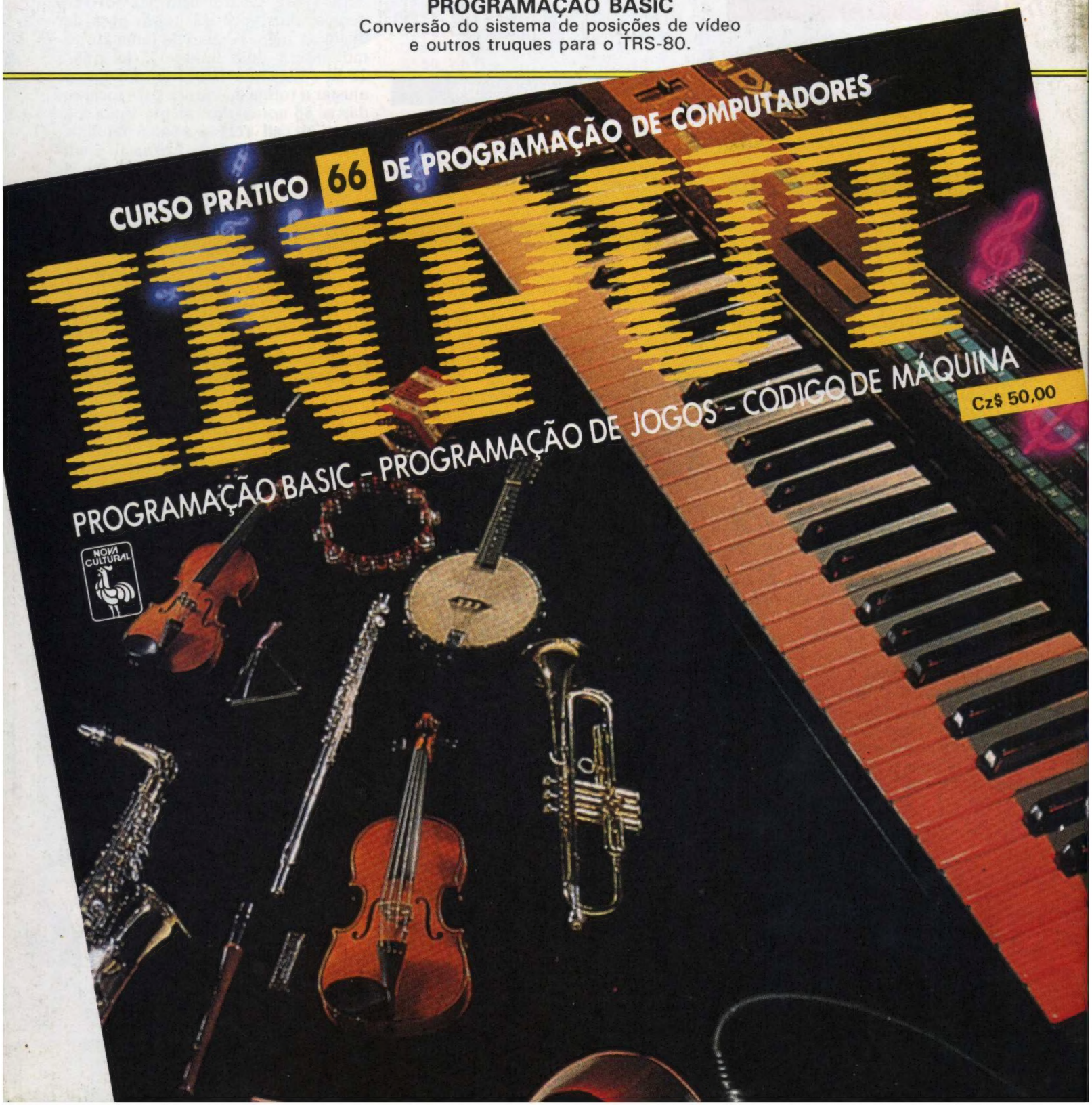


NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC
Quebra-cabeças no micro. Equações. Truques matemáticos.

PERIFÉRICOS
Computadores que ouvem. Unidades de reconhecimento da fala. Tipos de fala. Funcionamento e usos.

LINGUAGENS
LOGO e pseudo-LOGO. A tartaruga. Programação.

PERIFÉRICOS
Sintetizadores. Máquinas de percussão. A interface MIDI.

PROGRAMAÇÃO BASIC
Conversão do sistema de posições de vídeo e outros truques para o TRS-80.